

#2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of)
SUZUKI et al.)
Application Number: To Be Assigned)
Filed: Concurrently Herewith)
For: APPARATUS FOR SYSTEM DECODER AND METHOD)
FOR ERROR CORRECTION OF PACKET DATA)

11000 U.S. PTO
09/940576
08/29/01

Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231

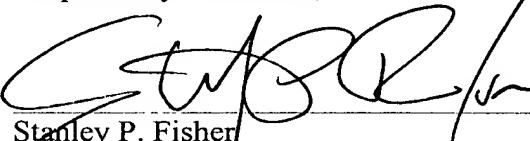
**REQUEST FOR PRIORITY
UNDER 35 U.S.C. § 119
AND THE INTERNATIONAL CONVENTION**

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of July 10, 2001, the filing date of the corresponding Japanese patent application 2001-208680.

The certified copy of corresponding Japanese patent application 2001-208680 is being submitted herewith. Acknowledgment of receipt of the certified copies is respectfully requested in due course.

Respectfully submitted,



Stanley P. Fisher
Registration Number 24,344

REED SMITH HAZEL & THOMAS LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200
August 29, 2001

JUAN CARLOS A. MARQUEZ
Registration No. 34,072

日本国特許庁
JAPAN PATENT OFFICE

J1000 U.S. PTO
09/940576
08/29/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出願年月日

Date of Application:

2001年 7月10日

出願番号

Application Number:

特願2001-208680

出願人

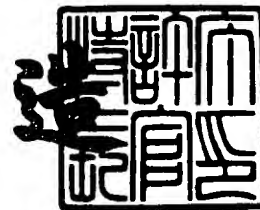
Applicant(s):

株式会社日立製作所

2001年 8月10日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造



出証番号 出証特2001-3071040

【書類名】 特許願

【整理番号】 H01008091A

【あて先】 特許庁長官 殿

【国際特許分類】 H03M 13/00

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 280 番地 株式会社日立製作所中央研究所内

【氏名】 鈴木 芳典

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 280 番地 株式会社日立製作所中央研究所内

【氏名】 横山 徹

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 280 番地 株式会社日立製作所中央研究所内

【氏名】 木村 淳一

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社 日立製作所

【代理人】

【識別番号】 100075096

【弁理士】

【氏名又は名称】 作田 康夫

【電話番号】 03-3212-1111

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 システムデコーダ装置及びパケットデータの修正方法

【特許請求の範囲】

【請求項 1】

入力した複数のパケットデータからペイロード部を取り出すパケット処理部と、到着していないパケットデータ及び伝送エラーを伴うパケットデータのシーケンス番号を検出し、前記検出されたシーケンス番号に対応する到着していないパケット及び伝送エラーを伴うパケットのペイロード部の代わりにエラー検出用のユニークワードを出力するパケットロス検出部と、前記ペイロード部と前記ユニークワードとをアプリケーションデコーダが再生する順序に入れ替えて出力するメディアデータ再構成部とを備えることを特徴とするシステムデコーダ装置。

【請求項 2】

前記ユニークワードは、アプリケーションデコーダの仕様に含まれていないデータ列であり、アプリケーションデコーダがエラーとして判定するデータ列であることを特徴とする請求項 1 記載のシステムデコーダ装置。

【請求項 3】

前記ユニークワードは、アプリケーションデコーダとの間で予めエラー通知用データとして定めておいたデータ列であることを特徴とする請求項 1 記載のシステムデコーダ装置。

【請求項 4】

前記パケットロス検出部からの入力を受けて、前記メディアデータ再構成部から出力されるメディアデータと同サイズの誤り通知データを出力する誤り通知出力部とを備えることを特徴とする請求項 1 記載のシステムデコーダ装置。

【請求項 5】

前記パケットロス検出部は、前記到着していないパケットデータ及び伝送エラーを伴うパケットデータのシーケンス番号を検出する際に、当該パケットのペイロードタイプを同時に検出し、ペイロードタイプに決定されるメディアデコーダの種類毎に異なったユニークワードを出力することを特徴とする請求項 1 記載のシステムデコーダ装置。

【請求項 6】

前記パケットロス検出部は、前記到着していないパケットデータ及び伝送エラーを伴うパケットデータのシーケンス番号を検出する際に、当該パケットのペイロードタイプを同時に検出し、前記誤り通知出力部は、該ペイロードタイプによりペイロードタイプに決定されるメディアデコーダの種類毎に異なった誤り通知データを出力することを特徴とする請求項 1 記載のシステムデコーダ装置。

【請求項 7】

入力した複数のパケットデータからペイロード部を取り出すパケット処理部と、到着していないパケットデータ及び伝送エラーを伴うパケットデータの再送を行うか否かの再送判定を行い結果を出力する再送判定部と、到着していないパケットデータ及び伝送エラーを伴うパケットデータのシーケンス番号を検出し前記再送判定部に該シーケンス番号を出力し、前記再送判定部の通知結果から到着していないパケット及び伝送エラーを伴うパケットのペイロード部の代わりにエラー検出用のユニークワードを出力するパケットロス検出部と、前記ペイロード部と前記ユニークワードとをアプリケーションデコーダが再生する順序に入れ替えて出力するメディアデータ再構成部とを備えることを特徴とするシステムデコーダ装置。

【請求項 8】

前記再送判定部は、アプリケーションデコーダでの処理状況に関連する情報を用いることにより到着していないパケットデータ及び伝送エラーを伴うパケットデータの再送の必要性を判定することを特徴とする請求項 7 記載のシステムデコーダ装置。

【請求項 9】

前記再送判定部は、前記到着していないパケットデータ及び伝送エラーを伴うパケットデータの重要性に基づき、該到着していないパケットデータ及び伝送エラーを伴うパケットデータの再送の必要性を判定することを特徴とする請求項 7 記載のシステムデコーダ装置。

【請求項 10】

入力した複数のパケットデータからペイロード部を取り出し、到着していない

パケットデータ及び伝送エラーを伴うパケットデータのシーケンス番号を検出し、前記到着していないパケット及び伝送エラーを伴うパケットのペイロード部の代わりにエラー検出用のユニークワードを出力し、前記ペイロード部と前記ユニークワードとをアプリケーションデコーダが再生する順序に入れ替えた信号を生成し、前記信号からユニークワードを検出し修正をおこなうパケットデータの修正方法。

【請求項 11】

前記修正をバイトアライン処理を含む修正手段によっておこなうことを特徴とする請求項 10 記載のパケットデータの修正方法。

【請求項 12】

前記信号がブロック単位の符号化アルゴリズムにて生成されたビデオデータである場合には、データの修正に、符号化ブロックのデータに含まれる符号化タイプの情報を変更する処理を含むことを特徴とする請求項 10 記載のパケットデータの修正方法。

【請求項 13】

前記信号がビデオデータである場合、データの修正に、フレーム画像のデータに含まれるタイムスタンプ情報を変更する処理を含むことを特徴とする請求項 10 記載のパケットデータの修正方法。

【請求項 14】

前記信号がブロック単位の符号化アルゴリズムにて生成されたビデオデータである場合、データの修正に、符号化ブロックを符号化データの無いブロックに変更すると共に当該ブロックに関連する符号化データの有無を示すフラグを変更する処理を含むことを特徴とする請求項 10 記載のパケットデータの修正方法。

【請求項 15】

前記信号が複数の符号化ブロックを 1 個のビデオパケットとして扱うことが可能なブロック単位の符号化アルゴリズムにて生成されたビデオデータである場合、データの修正に、ビデオパケットのヘッダ部に含まれるブロック位置情報を変更する処理を含むことを特徴とする請求項 10 記載のパケットデータの修正方法。

【請求項 1 6】

前記信号が複数の符号化ブロックを 1 個のビデオパケットとして扱うことが可能なブロック単位の符号化アルゴリズムにて生成されたビデオデータであ留場合、データの修正に、ビデオパケット内にて処理すべき符号化ブロックの数を変更する処理を含むことを特徴とする請求項 1 0 記載のパケットデータの修正方法。

【発明の詳細な説明】

【0 0 0 1】

【発明の属する技術分野】

本発明は、デジタルメディアデータの無線通信動画像符号化技術に関するものであり、特に、パケットロスが生じた場合の受信データ処理に関するものである。

【0 0 0 2】

【従来の技術】

次世代無線通信サービスとして、データ圧縮され、配信サーバに蓄積されているデジタルメディアデータファイルを、携帯電話のような無線携帯端末にて受信・再生することが想定されている。このような無線通信サービスで、パケットロス（伝送中のパケットデータの損失）が生じた時の通信の形態としては、以下の 2 点が想定されている。

- 1) 損失したパケット(データの単位、データのかたまりの総称)の再送を行わないコネクションレス型のトランスポート層プロトコル(例えば、UDP: User Datagram Protocol, RTP: Realtime Transport Protocol)を使用し、データを受信しながら再生するストリーミング、
- 2) 損失したパケットの再送を可能とするコネクション型のトランスポート層プロトコル(例えば、TCP: Transmission Control Protocol、全てのメッセージに対して、エラー検出や応答確認処理を行うため、信頼性を要求するアプリケーションに適している。但し、伝送路の状態が悪いと、頻繁な確認応答や再送要求により、スループット低下が起こりやすい)を使用し、全データを受信し、一旦蓄積してから再生処理を開始するダウンロード。(UDP, RTP, TCPの詳細は後述する)

ここで、トランスポート層とは、異機種間通信を可能にするネットワークアーキテクチャの標準参照モデル(OSI: Open Systems Interconnection, 開放型システム相互接続、通信回線に近いところから物理層-データリンク層-ネットワーク層-トランスポート層)において、「データを確実に相手に届ける」役割を果たすレベルのこと。「アドレスの管理や経路の選択を行う」役割を果たすIP(Internet Protocol)はネットワーク層にあたる。上記2通りのサービス形態のうち、ダウンロードでは、データ損失を伴う全てのパケットに対して再送要求をかけるため、受信データに伝送エラーが含まれることはない。一方、ストリーミングサービスでは、データ損失を伴うパケットが存在しても再送されないため、受信データに伝送エラーが含まれる可能性がある。

【0003】

ここで、具体的な事例を示す。図2は携帯電話のシステム構成を示す図であり、図3は、送信されるパケットを表したものである。この携帯電話端末200に向けて、図3に示すトランスポート層のパケット列(1種類のメディアデータ)が送信された場合を考える。TCP、UDP-RTPに関わらずトランスポート層の1パケットは、トランスポート層のヘッダ部(411, 421, 431, 441, 451)とペイロード部(412, 422, 432, 442, 452)にて構成されている。この例では、ペイロード部は実際のメディアデータである。ヘッダ部には、ペイロード部のデータ量(バイト単位)、パケットの通し番号(シーケンス番号)、ペイロードタイプ(RTPの場合のみ、TCPでは1度に複数タイプのデータを扱うことはない。ビデオデータ、オーディオデータの識別をこのデータにて行う)ならびにタイムスタンプ(RTPの場合のみ)等が書かれている。まず、アンテナ201から受信した無線信号は、無線インタフェース部202にて電気的に変換され、データリンク層プロトコルのパケット203の形式でシステムデコーダ205に入力される。システムデコーダ205では、入力されたデータリンク層プロトコルのパケットがネットワーク層のパケット、トランスポート層のパケットへと順にプロトコル変換され、トランスポート層のペイロード部だけがメモリ207に出力される。また、システムデコーダでは、パケット内のエラー検出処理や、各パケットにおけるペイロードタイプの識別(RTPの場合のみ)等も行う。更に、

異なるペイロードタイプのメディアデータが同時に配信されている場合には、ペイロードタイプ別にメディアデータを構成し、共に互いに同期を取りながらメモリ207に出力される。

【0004】

トランスポート層のプロトコルがTCPである場合には、システムデコーダは、エラーなく到達した各パケットについて、応答確認情報204を無線インタフェース部202に返答する。これらの応答確認情報は、無線インタフェース部202にて無線信号に変換され、送信側に返信される。送信側では、送信したパケットの応答確認を待ってから次のパケットの送信処理を行うが(複数のパケットを並列に処理することも可能)、送信から一定時間が経過しても応答確認が到達しない場合には、そのパケットを携帯電話端末200に再送する。そのため、TCPではパケットロスが発生せず、図4の2060に示されるようなペイロード部のみから成るデータがペイロードデータ206としてメモリ207に出力される。ここで言うペイロードデータとは、一般的に、ダウンロードの場合には、画像・音声等をまとめたファイルフォーマットのデータとなり、具体的には、MPEG-4のMP4ファイルフォーマット、アップル社のQuick Time、Microsoft社のWMT(Window s Media Technology)などがある。

【0005】

トランスポート層のプロトコルがUDP-RTPである場合には、システムデコーダは、応答確認を行わないため、パケットロスは許容される。例えば、図3の第2パケット(421及び422)がシステムデコーダに到着しなかった場合、あるいは第2パケットに伝送エラーが含まれている場合には、図5の2061に示されるような第2パケットを除いたペイロード部のみから成るデータがペイロードデータ206としてメモリ207に出力される。ここで言うペイロードデータとは、ストリーミングの場合には、一般にビデオ符号化データの packets とオーディオ符号化データの packets が混成されたデータとなる。このように複数のメディアデータ(例えば、ビデオデータとオーディオデータ)が同時に配信されており、同時に再生する必要がある場合には、システムデコーダ205にて、RTP packets のヘッダ部に含まれているペイロードタイプによって各 packets が関連するメディア

デコーダ(例えば、ビデオデコーダ、オーディオデコーダ)の種類が解読され、異なる種類のメディアデータとしてメモリ207に出力する必要がある。

その後、メモリ207に蓄積されたデータは、ストリーミングの場合には順次、最終パケットが到達した時点でメディアデータ208としてアプリケーションデコーダ209に出力される。なお、本明細書では、アプリケーションデコーダ209を、ファイルフォーマットのファイルデコーダならびに、ビデオデコーダ、オーディオデコーダなどのメディアデコーダの総称とし、アプリケーションデコーダと示した場合には、内部の構成は限定しないことにする。アプリケーションデコーダでは、ロスパケットの誤り隠蔽処理を行いつつ、タイミングをはかって各メディアデータの復元・再生処理を行う。但し、メディアデコーダに含まれる誤り隠蔽処理は標準化されていないため、メディアデータ208にパケットロスが含まれる場合の再生画像の画質や音声品質は、誤り隠蔽処理の性能によって異なることになる。

【0006】

【発明が解決しようとする課題】

伝送中にパケットロスが発生した場合、アプリケーションデコーダに渡されるメディアデータは、ロスが発生したパケットのペイロードを省略した形式となる。一般にアプリケーションデコーダでは、適用標準デコード方式に含まれないデータの組み合わせを入力データ内に発見したときに伝送誤りが発生していると判断する。そのため、ロスしたパケットの前後のパケットを結合したデータが適用標準デコード方式に含まれる組み合わせとなっていた場合、伝送誤りの正確な位置を検出することができない。

【0007】

また、パケットロスを含むデータを高画質で再現するためには、アプリケーションデコーダが伝送誤りを検出し、エラーの影響が画質に現れないように制御する必要がある。このような誤り訂正処理はデコーダの処理量を増加させるため、製品価格や消費電力の関係で高速なプロセッサを使用できないアプリケーションでは、再生処理速度や画像サイズの面で要求を十分に満たせないことになる。

【0008】

【課題を解決するための手段】

受信データのパケットロス部分に、アプリケーションデコーダが明示的に通信エラーが発生したと判定できるデータ列をシステムデコーダにて埋め込む。具体的には、アプリケーションデコーダの仕様・規格に含まれないデータ列を埋め込みデータとして利用する。これにより、アプリケーションデコーダは、伝送誤りの正確な位置を検出することが可能となる。また、埋め込むデータ列をシステムデコーダとアプリケーションデコーダの間で予め決めておくことにより、より正確な誤り位置検出が可能となる。さらに、このエラー検出用埋め込みデータを利用して、誤りを含んだメディアデータをアプリケーションデコーダの仕様・規格に準拠したデータに変換することで、アプリケーションデコーダにおける誤り訂正処理の負担を軽減することが可能となる。

【0009】

【発明の実施の形態】

まず、図2、図38、図39、図6ならびに図7にて、パケットロスの発生を検出する仕組みを考える。図6は、図2に示すシステムデコーダ205の内部構成を示したものである。

入力されたパケットデータは、まず、パケット処理部301にてトランスポート層のパケットとへ変換される。

トランスポート層のプロトコルがUDP-RTPの場合には、各UDPパケットのヘッダ部に含まれているエラー検出情報を解析し、受信パケットがエラーを含んでいる場合には、パケット処理部301にてそのパケットを破棄する。RTPパケットは、UDPパケットのペイロード部にあたり、ペイロードデータと、シーケンス番号ならびにペイロードタイプ情報を含むRTPヘッダにて構成されている。そのため、後に送信されたパケットが先に受信側に到着した場合でも、システムデコーダのメディアデータ再構成部308にて送信された順序に入れ替えることが可能となる。そこで、受信したUDPパケットがエラーを含んでいない場合には、RTPパケットを解析し、ペイロードデータ307と、タイムスタンプ・シーケンス番号・ペイロードタイプ801をメディアデータ再構成部308に出力する。この際、タイムスタンプ・シーケンス番号・ペイロードタイプ801をパケットロス検出部8

02にも出力する。これにより、パケットロス検出部802では、シーケンス番号とペイロードタイプを解析することで、到着していないパケットデータならびに伝送エラーを伴うパケットデータをペイロードタイプ別に検出することが可能となる。具体的には、メディアデータ再構成部308から出力すべきタイミングになっても、シーケンス番号とペイロードタイプがパケットロス検出部802に届いていないパケットを、ロスパケットであると判定する。なお、ペイロードデータ307と、タイムスタンプ・シーケンス番号・ペイロードタイプ801を受け取ったメディアデータ再構成部308では、各RTPパケットのシーケンス番号とペイロードタイプに従って、ペイロードタイプ毎にパケット列の順序の入れ替える操作を行う。例えば、図7の(a)のように、「あるペイロードタイプにおける2番目のパケット(421と422)が未着信」で、「通信に使用したネットワークの状況により、そのペイロードタイプのパケットが、第1パケット、第4パケット、第5パケット、第3パケットの順序で携帯端末側に着信した」場合を想定する。この場合、パケット処理部301は、到着したパケットを到着順(411と412, 441と442, 451と452, 431と432)に処理し、図7の(b)に示すように412, 442, 452, 432からなるペイロードデータ列307とRTPヘッダ部のタイムスタンプ・シーケンス番号・ペイロードタイプ列801(図中では、シーケンス番号のみを表示している)に分けてメディアデータ再構成部308に対して出力する。メディアデータ再構成部308では、各RTPパケットのシーケンス番号ならびにペイロードタイプに従って、ペイロードタイプ毎にパケット列の順序の入れ替える操作を行う。図7の例では、図5に示すようなペイロードデータ2061が、システムデコーダとアプリケーションとのインタフェースの役割を果たすメディアデータ出力部310を通してメモリ207に出力される。メディアデータ再構成部308がメディアデータ出力部310に各ペイロードデータを出力するタイミングは、一般に各RTPパケットに含まれるタイムスタンプによって管理されるが、アプリケーションデコーダの再生状況を考慮しながら制御することも可能である。

図38は携帯電話端末の別の実施例を示す図で、2種類のメディアデータ(ビデオデータとオーディオデータ)がRTPにてストリーミング配信された場合について

示している。図38では、図2に示した携帯電話端末200と、システムデコーダからアプリケーションデコーダへのデータの流れの点で異なる。

この場合には、RTPパケットヘッダ部のペイロードタイプは、ビデオデータあるいはオーディオデータを示している。システムデコーダ2051は受信した各RTPパケットのヘッダ部に記載されているペイロードタイプとシーケンス番号を調査し、ビデオデータとオーディオデータに分け、それぞれシーケンス番号順にペイロードデータ2061をメモリ2071に出力する。この際の各ペイロードの出力タイミングは、RTPパケットヘッダ部のタイムスタンプを参考に、再生時にビデオとオーディオ間で互いに同期が取れるように決められる。メモリ2071に蓄積された各メディアデータ2081は、それぞれ再生単位のデータ量に達した時点で、アプリケーションデコーダ2091内のビデオデコーダならびにオーディオデコーダに入力され、再生処理される。

【0010】

一方、トランスポート層のプロトコルがTCPである場合には、各TCPパケットのヘッダ部に含まれているエラー検出情報を解析し、エラー検出結果302を再送判定部304に渡す。その際、受信パケットがエラーを含んでいる場合には、そのパケットを破棄する。再送判定部304では、エラー検出結果302が「エラーなし」を示している場合には、応答確認情報305を応答確認部306に通知する。この応答確認情報305は、応答確認部306にて無線信号204に変換され、無線インタフェース部202を通して送信側に通知される。送信側では、送信から一定時間が経過しても応答確認が到達しないパケットを携帯電話端末200に再送する。プロトコルがTCPである場合には、送信側のパケット送信は応答確認の到着状況を考慮して行われており、基本的に不正なパケットは発生しない。しかしながら、最近では、ダウンロードの信頼性とストリーミングのリアルタイム性の中間にあたる性能を持つサービスとして、擬似ストリーミングも提案されている。この擬似ストリーミングは、「ダウンロードと同様に信頼性の高いコネクション型のトランスポート層プロトコルを使用するが、メディアデータを数個のファイルに分けて送信することにより、再生に必要な全データの受信を待たずに受信したファイル内のメディアデータから適宜再生する」というものであ

る。ダウンロードは、通信の信頼性は高いが、ファイルが完全に受信し終わるまで再生処理を開始することができない。そのため、携帯電話のように大きなメモリを組み込めない端末では、再生時間の長いメディアデータは受信できないという問題がある。そこで、送信側にて、予めメディアデータを分割しておき、到着したファイルから再生処理を行うことで、処理が終わったファイルをメモリから削除していくという方法が有効と成る。しかしながら、無線通信にて擬似ストリーミングを行う場合には、パケットの再送が頻繁に発生すると、あるファイルの再生が終了しても、次のファイルが到着せず、再生を一時停止しなければならないという問題が発生する。そこで、擬似ストリーミングでは、TCPのように送信側とのコネクションを持つ通信方式であっても、ある程度のパケットエラーを許容する仕組みを取り入れる必要が生じてくる。なお、このパケットエラーを許容する方法は、ユーザ指令からの遅延等を考えた場合、ダウンロードであっても有効となる。

【 0 0 1 1 】

TCPにてパケットロスを許容する場合のパケットロス検出方法は、パケットロスを許容する方法そのものである。パケットロスを許容するには、送信側あるいは端末側のいずれかで応答確認手順を制御すればよい。送信側で制御するには、一定時間が経過しても応答確認が到達しない場合や再送回数が一定回数を越えた場合などの条件で、応答確認の到達をあきらめることで実現される。この場合、パケットロスの発生は、TCPパケットヘッダ部のシーケンス番号列にて確認できる(TCPでは、同時に応答確認を行うパケット数が端末側に通知されるため、その条件より大きなシーケンス番号を持つパケットが着信した時点でパケットロスの発生が判断できる)。しかし、この方法では端末側のアプリケーションデコードにおける再生処理の進行状況を応答確認手順の制御に反映することができないため、各ファイルの処理時間の推定値と実値と間にズレがあった場合には、再生処理を一時停止せざるを得ない状況が生じる可能性がある。

そこで、次に端末側で応答確認処理を制御する方法について考える。具体的には、図6の再送判定部304とパケットロス検出部802を利用する。パケット処理部301は、各TCPパケットのヘッダ部に含まれているエラー検出情報を解析

し、エラー検出結果 3 0 2 を再送判定部 3 0 4 に渡す。この際、受信パケットがエラーを含んでいる場合には、そのパケットを破棄する。一方、受信パケットがエラーを含んでいない場合には、TCPパケットを解析し、ペイロードデータ 3 0 7 と、シーケンス番号 8 0 1 (TCPには、ペイロードタイプは存在しない)をメディアデータ再構成部 3 0 8 に出力する。同時にシーケンス番号 8 0 1 をパケットロス検出部 8 0 2 に出力する。これにより、パケットロス検出部 8 0 2 では、シーケンス番号を解析することで、受信していないパケットを検出することが可能となる。具体的には、パケット処理部 3 0 1 からシーケンス番号が届いていないパケットを、受信していないパケットであると判定する。パケットロス検出部 8 0 2 では、前後のパケットが受信された後、規定時間(システムデコーダにて制御)が経過しても受信されないパケットをロスパケットとし、そのシーケンス番号 8 0 7 を再送判定部 3 0 4 に通知する。再送判定部 3 0 4 では、ロスパケットの再送により再生処理に支障があるか否かを判断する。そして再送を待つと再生処理遅延の面で問題があると判断した場合には、応答確認情報 3 0 5 を応答確認部 3 0 6 に通知するとともに、再送を行わないことを示す再送判定結果・シーケンス番号 3 0 3 をパケットロス検出部 8 0 2 に返答する。再送を待っても再生処理が滞りなく進行すると判断できる場合には、再送を行うことを示す再送判定結果・シーケンス番号 3 0 3 をパケットロス検出部 8 0 2 に返答する。この返答結果から、パケットロス検出部 8 0 2 は、メディアデータ再構成部 3 0 8 にて受信できていないパケットが、再送待ちのパケット(再送を行うことを示す再送判定結果が通知されている)であるか、パケットロスとして許容されたパケットであるか(再送を行わないことを示す再送判定結果が通知されている)が判断できる。そして、再送待ちのパケットについては、規定時間(システムデコーダにて制御)経過しても受信しない場合には、再度、再送判定部 3 0 4 に、シーケンス番号 8 0 7 を通知する。再送判定部 3 0 4 では、再送回数も考慮に入れて、再送要求の可能性について判定を行う。なお、再送判定部 3 0 4 がロスパケットと判定した後で、受信したパケットについては、通常通り処理することで情報の書き換えることも可能となる。このような手順で、ペイロードデータ 3 0 7 とシーケンス番号を受け取ったメディアデータ再構成部 3 0 8 では、各TCPパケットのシーケ

ス番号に従って、受信パケットの出力順序を入れ替える操作が行われる。入れ替え後、メディアデータ再構成部 3 0 8 がメディアデータ出力部 3 1 0 にペイロードデータを出力するタイミングは、一般に、一定の時間間隔で制御されるが、アプリケーションデコーダの再生状況を考慮しながら制御することも可能である。図 3 9 は携帯電話端末のさらなる実施例を示す図で、2 種類のメディアデータ(ビデオデータとオーディオデータ)を 1 つのファイルに統合・蓄積した MP4 ファイルフォーマットデータが TCP にて擬似ストリーミング配信された場合について示している。図 3 9 では、図 2 に示した携帯電話端末 2 0 0 と、システムデコーダからアプリケーションデコーダへのデータの流れが異なっている。

システムデコーダ 2 0 5 2 は TCP パケットヘッダ部のシーケンス番号を調査し、シーケンス番号順にペイロードデータ 2 0 6 2 をメモリ 2 0 7 2 に出力する。この際の各ペイロードの出力タイミングは、アプリケーションデコーダにおけるメディアデータの処理状況を考慮して決める。また、システムデコーダにおけるロスパケットの再送制御にもアプリケーションデコーダでの処理状況を反映する。メモリ 2 0 7 2 に蓄積されたメディアデータ 2 0 8 2 は、1 つのファイルフォーマットデータとして成立した時点で、アプリケーションデコーダ 2 0 9 2 内のファイルデコーダに入力される。ファイルデコーダでは、ファイルヘッダの情報からファイル内の各メディアデータ(ここでは、ビデオデータとオーディオデータが含まれていることを想定)を取りだし、それぞれビデオデコーダならびにオーディオデコーダに入力される。

図 4 0 に携帯電話端末の別の実施例を、図 4 1 にシステムデコーダの別の実施例を示し、アプリケーションデコーダでの処理状況を考慮して、再送判定部 3 0 4 にてロスパケットならびにエラーパケットの再送要求判定を行うためのデータの流れを説明する。図 4 0 に示したデータ 2 1 0 ならびに 2 1 1 が処理状況関連情報にあたる。データ 2 1 1 としては、メモリ内の未処理データの残量、データ 2 1 0 としては、再生遅延の発生状況、フレームレートならびにビットレートなどが含まれる。これらのデータは、図 4 1 の再送判定部 3 0 4 に送られる。再送判定部 3 0 4 では、データ 2 1 1 により得られるデータ残量と、データ 2 1 0 から得られるデータの消費速度に関する情報、さらに再送パケットが到達するにかか

る時間の推定値(それまでのデータ到達状況から推測する)、そのパケットの重要性(例えば、ファイルフォーマットデータのヘッダ部やビデオデータのシーケンスヘッダ部を失うとデータ全体の再生が困難となる)等から、パケット再送待ちがアプリケーションデコーダに与える影響を推定する。

【 0 0 1 2 】

TCPにおいてパケットロスを制御する方法としては、上記以外に、送信側と受信端末側の双方で制御することも可能である。この場合には、応答確認情報以外に、再送中止要求や再送要求を端末側から送る仕組みを作ることにより高速で緻密な制御が可能となる。

【 0 0 1 3 】

以上説明したTCPを利用した擬似ストリーミングでは、1回の通信で1種類のメディアデータしか扱えないという問題がある。そのため、メディアデータを分割した各ファイルにはファイルヘッダをつけなければならず、通信容量の増大を招く可能性がある。そこで、擬似ストリーミングでは、TCP-RTPという通信手段も有効である。ここでは、端末側で応答確認処理を制御する方法についてTCP-RTPの仕組みを図6にて説明する。パケット処理部301は、各TCPパケットのヘッダ部に含まれているエラー検出情報を解析し、エラー検出結果302を再送判定部304に渡す。この際、受信パケットがエラーを含んでいる場合には、そのパケットを破棄する。一方、受信パケットがエラーを含んでいない場合には、TCPパケットヘッダとTCPペイロードに含まれるRTPヘッダを解析し、RTPペイロードデータ307と、TCPシーケンス番号・RTPタイムスタンプ・RTPシーケンス番号・RTPペイロードタイプ801をメディアデータ再構成部308に出力する。同時にTCPシーケンス番号・RTPタイムスタンプ・RTPシーケンス番号・RTPペイロードタイプ801をパケットロス検出部802に出力する。これにより、パケットロス検出部802では、TCPシーケンス番号、RTPペイロードタイプ、RTPシーケンス番号を解析することで、受信していないRTPパケットとそのペイロードタイプを検出することが可能となる。具体的には、パケット処理部301からTCPシーケンス番号が届いていないパケットを、受信していないパケットであると判定し、RTPペイロードタイプとRTPシーケンス番号からその種別を判定する。パケッ

トロス検出部 8 0 2 では、前後の TCP パケットが受信された後、規定時間(システムデコーダにて制御)が経過しても受信されない TCP パケットをロスパケットとし、その TCP シーケンス番号 8 0 7 を再送判定部 3 0 4 に通知する。再送判定部 3 0 4 では、ロスパケットの再送により再生処理に支障があるか否かを判断する。そして再送を待つと再生処理遅延の面で問題があると判断した場合には、応答確認情報 3 0 5 を応答確認部 3 0 6 に通知するとともに、再送を行わないことを示す再送判定結果・TCP シーケンス番号 3 0 3 をパケットロス検出部 8 0 2 に返答する。再送を待っても再生処理が滞りなく進行すると判断できる場合には、再送を行うことを示す再送判定結果・TCP シーケンス番号 3 0 3 をパケットロス検出部 8 0 2 に返答する。この返答結果から、パケットロス検出部 8 0 2 は、メディアデータ再構成部 3 0 8 にて受信できていない TCP パケットが、再送待ちのパケット(再送を行うことを示す再送判定結果が通知されている)であるか、パケットロスとして許容されたパケットであるか(再送を行わないことを示す再送判定結果が通知されている)が判断できる。そして、再送待ちのパケットについては、規定時間(システムデコーダにて制御)経過しても受信しない場合には、再度、再送判定部 3 0 4 に、TCP シーケンス番号 8 0 7 を通知する。再送判定部 3 0 4 では、再送回数も考慮に入れて、再送要求の可能性について判定を行う。なお、再送判定部 3 0 4 がロスパケットと判定した後で、受信したパケットについては、通常通り処理することで情報を書き換えることも可能となる。このような手順で、ペイロードデータ 3 0 7 と TCP シーケンス番号・RTP タイムスタンプ・RTP シーケンス番号・RTP ペイロードタイプを受け取ったメディアデータ再構成部 3 0 8 では、各 RTP パケットのシーケンス番号ならびにペイロードタイプに従って、ペイロードタイプ毎にパケット列の順序の入れ替える操作が行われる。メディアデータ再構成部 3 0 8 がメディアデータ出力部 3 1 0 に各ペイロードデータを出力するタイミングについては、各 RTP パケットに含まれるタイムスタンプとアプリケーションデコーダの再生状況を考慮しながら制御する方法が有効である。システムデコーダから出力された各ペイロードの再生方法は、例えば、図 3 8 にて説明される。2 種類のメディアデータ(ビデオデータとオーディオデータ)が TCP-RT P にて擬似ストリーミング配信された場合では、RTP パケットヘッダ部のペイロー

ドタイプは、ビデオデータあるいはオーディオデータを示している。システムデコーダ2051は受信した各RTPパケットのヘッダ部に記載されているペイロードタイプとシーケンス番号を調査し、ビデオデータとオーディオデータに分け、それぞれシーケンス番号順にペイロードデータ2061をメモリ2071に出力する。この際の各ペイロードの出力タイミングは、RTPパケットヘッダ部のタイムスタンプとアプリケーションデコーダにおけるメディアデータの処理状況を考慮して決められる。また、システムデコーダにおけるロスパケットの再送制御にもアプリケーションデコーダでの処理状況を反映する。システムデコーダにおけるロスパケットの再送制御は、TCPの場合と同じく図40ならびに図41にて説明できる。図40に示したデータ210ならびに211が処理状況関連情報にあたる。データ211としては、メモリ内の各ペイロードタイプにおける未処理データの残量、データ210としては、再生遅延の発生状況、フレームレートならびにビットレートなどが含まれる。これらのデータは、図41の再送判定部304に送られる。再送判定部304では、データ211により得られるデータ残量と、データ210から得られるデータの消費速度に関する情報、さらに再送パケットが到達するにかかる時間の推定値(それまでのデータ到達状況から推測する)、ペイロードタイプの重要性(基本的にオーディオ再生が途切れることは許されない)、ロスパケットのペイロードタイプ割合、各ペイロードタイプのタイムスタンプ等から、パケット再送待ちがアプリケーションデコーダに与える影響を推定する。そして、メモリ207に蓄積された各メディアデータ208は、「擬似ストリーミングにおけるTCP通信の伝送速度を考慮した受信から再生開始までの待ち時間」を考慮して規定されたデータ量に達した時点で、順にアプリケーションデコーダ209内のビデオデコーダならびにオーディオデコーダに入力され、再生処理される。

【0014】

ここで、パケット処理部301の動作について触れておく。図8は、IS-95規格無線端末のパケット処理部の詳細を示したものである。まず、無線信号は無線インタフェース部202にてデータリンク層制御プロトコルであるRLP (Radio Ling Protocol) のフレームデータ902(フレームとパケットはともにデー

タ単位のこと)に変換され、RLP解析部に入力される。図11にRLPフレームのデータフォーマットを示す。RLPはCDMAデータに対応した通信プロトコルであり、データリンク層のプロトコルとのインタフェースを保つ役割を果たす。RLP解析部903では、RLPヘッダ情報に基づいて、複数のRLPペイロードデータ904を組み合わせることで、PPP (Point to Point Protocol) に対応したフレームデータ9041を構成し、PPP解析部905に出力する。図10にPPPフレームのデータフォーマットを示す。PPPは、データリンク層のプロトコルの一種でありシリアル回線にて使用される。データリンク層は、主に「物理的な通信を確立する」役割と「データが通信路を通過している間に発生したエラーの検出を行う」役割を果たす。PPP解析部905では、使用されているネットワーク層プロトコルの判別（上位プロトコルデータを解析）、データが通信路を通過している間に発生したエラーの検出（誤り検査データを解析）、ならびに複数ペイロードデータ906を組み合わせたIPパケットデータ9061の構成の処理を行う。なお、ここでは、上位プロトコルであるネットワーク層のプロトコルとしてIPv4が採用されているものとして記載する（一般的に知られているIPとしては、他に、IPv6、AppleTalk等があり、勿論本発明に適用可能である）。図9にIPv4パケットのデータフォーマットを示す。IPv4パケットデータ906は、IP解析部907に入力される。IP解析部では、使用されているトランスポート層プロトコルの判別（上位プロトコルデータを解析）やヘッダ部のエラーの検出（チェックサムデータを解析）を行うと共に、複数のIPペイロードデータ908を組み合わせることで、上位プロトコルのパケットを構成する。そして、上位パケットがTCPであった場合には、TCP解析部910にTCPパケット9080を出力し、UDPであった場合には、UDP解析部911にUDPパケット9081を出力する。図12にTCPパケットのデータフォーマット、図13にUDPパケットのデータフォーマットを示す。TCPとUDPの使い分けは、伝送データの性質により行われ、一般的には、伝送速度よりも信頼性が重要なデータではTCP、リアルタイム性が重要な場合にはUDPが用いられる。TCP解析部910では、TCPパケットのヘッダ部に含まれるチェックサムを解析することでエラー検出処理が行われ、エラー検出結果302が再送判定部304に渡される（図6に示したエラー検出結果302ならびに再送判定部304と同じ）

。エラー検出結果がエラーなしを示している場合には、ペイロードデータ307がメディアデータ再構成部308に、送信元シーケンス番号801がパケット検出部802とメディアデータ再構成部308に渡される。一方、UDP解析部では、UDPペイロードデータを、RTPパケット9121としてRTP解析部913に出力する。UDPパケットのチェックサムはオプション機能であり、エラー検出も必須の作業ではないが、安定した通信を提供するため、使用することが推奨されている。RTP解析部913では、ペイロード307をメディアデータ再構成部308に、タイムスタンプ・シーケンス番号・ペイロードタイプ801をパケット検出部802とメディアデータ再構成部308に振り分ける。

【0015】

図42は、図8のTCP部の処理をTCP-RTPに置き換えた図である。図8との相違点は、TCP-RTPの部分である。TCP解析部では、TCPパケットのヘッダ部に含まれるチェックサムを解析することでエラー検出処理が行われ、エラー検出結果302が再送判定部304に渡される(図6に示したエラー検出結果302ならびに再送判定部304と同じ)。エラー検出結果がエラーなしを示している場合には、TCPペイロードデータを、RTPペイロードデータ9120としてRTP解析部に渡すとともに、TCPヘッダ部のTCPシーケンス番号9122をパケットロス部とメディアデータ再構成部に渡す。RTP解析部913では、RTPペイロード307をメディアデータ再構成部308に、RTPヘッダのタイムスタンプ、シーケンス番号・ペイロードタイプをTCPシーケンス番号と共にパケット検出部802とメディアデータ再構成部308に振り分ける。

【0016】

次に図6のパケットロス検出部802にて取得したパケットロス情報を用いて、パケットロス位置をアプリケーションデコーダに明示的に通知する方法を考える。図5に示されるように、パケットロスが発生した場合には、失ったパケット部分のデータが省略されてアプリケーションデコーダに渡されることになるが、パケットロスの情報はアプリケーションデコーダには通知されない。そのため、アプリケーションデコーダがパケットロスの発生ならびにその発生位置を加味して再生処理を行えるか否かは、アプリケーションデコーダの誤り検出能力に委ね

られることになる。例えば、ロスしたパケットの前後をつなげた時のデータ列が、偶然、アプリケーションデコーダの仕様に含まれていた場合には、パケットロスの検出が遅れる原因となることも考えられる。本発明では、アプリケーションデコーダがパケットロスの発生位置を正確に判定できるようにするために、パケットロス部分にアプリケーションデコーダが理解できない符号化列を埋め込む仕組みを考える。具体的には、アプリケーションデコーダにおけるエラー検出を容易にするために、デコーダが理解できないデータ列(エラー検出用埋め込みデータ)をパケットロス部分に埋め込む。アプリケーションデコーダは理解できないデータ列をエラーと判断するため、パケットロスの発生位置をほぼ正確に検出可能となる。これにより、デコーダがエラーデータを正常データとして扱ってしまうようなエラー検出の遅延が回避され、再生画像・音声の乱れが抑制できる。なお、ここで言うアプリケーションデコーダには、ファイルフォーマットのファイルデコーダや、ビデオデコーダ、オーディオデコーダ、シーン記述デコーダ(複数のオブジェクトの表示位置再生処理)等のメディアデコーダが含まれる。従って、アプリケーションデコーダが理解できないデータ列(エラー検出用埋め込みデータ)は、アプリケーションデコーダに含まれるデコーダの種類によって異なる。例えば、図39のようにアプリケーションデコーダがMP4ファイルのファイルデコーダとビデオ・オーディオのメディアデコーダで構成されている場合には、ファイルデコーダ、ビデオデコーダ、オーディオデコーダの全てが理解できないデータ列を「アプリケーションデコーダが理解できない符号化列(エラー検出用埋め込みデータ)」としてもよいし、ロスパケットがMP4ファイルのヘッダ部にあたる場合にはファイルデコーダのみが理解できない符号列、ビデオデータ部にあたる場合にはビデオデコーダのみが理解できない符号列、オーディオデータ部にあたる場合にはオーディオデコーダのみが理解できない符号列を埋め込むことで、アプリケーションデコーダ全体として理解できない符号列(エラー検出用埋め込みデータ)を定義してもよい。また、図38のようにアプリケーションデコーダがオーディオデコーダとビデオデコーダで構成される場合には、システムデコーダにて、受信パケットが、ビデオデータとオーディオデータに分けて処理されるため、ペイロードタイプ別に異なるエラー検出用埋め込みデータを用意し、

アプリケーションデコーダ全体として理解できない符号列(エラー検出用埋め込みデータ)を定義すればよい。

【0017】

エラー検出用データ列を埋め込む手順は次に述べるとおりである。前に述べたように、メディアデータ再構成部308まで到達していないパケットのシーケンス番号(ならびにペイロードタイプ)と再送処理状況(再送処理中/パケットロス)は、パケットロス検出部802にて認識されている。そこで、図6と図41に示すように、エラー検出用埋め込みデータ805をパケットロス検出部802にて作成し、パケットロスが発生したシーケンス番号(ならびにペイロードタイプ)806と共にメディアデコーダ再構成部308に通知する仕組みをシステムデコーダ内に設ける。そして、メディアデコーダ再構成部308は、ロスしたパケットのシーケンス番号(ならびにペイロードタイプ)806に対応する順番で、エラー検出用埋め込みデータ805を出力する。これにより、エラー検出用データの埋め込みが可能となる。ここでは、簡素化のため、パケットロスが発生した部分のメディアデータを復号するアプリケーションデコーダが、MPEG-4ビデオ規格であった場合、つまり図39で、RTPパケットのペイロードタイプがビデオであった場合を例として、埋め込みデータのタイプを考える。

図15は、図5のように第2番目のパケットをロスした場合のエラー検出用埋め込みデータの例である。MPEG-4ビデオ規格では2進数表示で24個以上の0を続けることは禁止されている。従って、埋め込みデータ8052は、MPEG-4ビデオ規格には含まれないデータ列であり、ビデオデコーダには理解できず、エラーが発生したと認識できる。しかしながら、この例のように、アプリケーションデコーダ(ここでは、ビデオデコーダ)との間で特に取り決めのないユニークワードをエラー検出用埋め込みデータとして使用した場合では、パケットロス位置の周辺で誤りを検出することは可能であるが、エラーデータと正常データの切れ目を正確に検出することは難しい。従って、既存の市販ビデオデコーダをそのまま利用し、システムデコーダを新規開発する場合のパケットロス位置検出方法としては有効であるが、十分な検出性能は得られない。

そこで図1に、アプリケーションデコーダとの間で予め決めておいたユニークワ

ード 8 0 5 1 を埋め込んでおく場合の MPEG-4 ビデオ規格の例を示す。MPEG-4 ビデオでは、2 3 個の 0 と 1 個の 1 の組み合わせをデータの切れ目を示すための識別コードとしており、その 2 4 ビットに続く 2 バイトのデータがその切れ目の種類を示すように設計されている。図 1 で使用している、1 6 進数の A 1 ならびに A 5 は、リザーブ用の識別コードとされており、現在の規格では使用されていない。従って、このようなユニークワードは MPEG-4 ビデオの標準デコーダでは理解できず、エラーと見なされることになる。そこで、このようなデータ列が予めエラー検出用埋め込みデータであることを、MPEG-4 ビデオデコーダとパケットロス検出部 8 0 2 が理解しておくことにより、ビデオデコーダにて正確なパケットロスの位置検出が実現でき、正常データとエラー検出用埋め込みデータの切れ目も明確となる。また、2 個のユニークワードを設定し、これらをエラー検出用埋め込みデータの最初と最後につけておく方法も有効である。この方法では、第 1 のユニークワードの数ビットを誤ってデコードしてしまったとしても、埋め込みデータ区間の途中でエラーが検出されるように細工しておけば(2 4 個以上の 0 を挟む)、第 2 のユニークワードを検索することにより正常データが始まる位置を正確に検出することが可能となる。また、一度処理したデータを再度検索し直すことが可能な場合には、少し前に戻ってから、第 1 のユニークワードを改めて検索することにより、パケットロスが始まる位置も正確に検出することが可能となる。

【 0 0 1 8 】

図 1 6 は、メディアデータのほかに、それと同サイズの誤り通知データを用意する方法である。誤り通知データにおける各バイナリーデータを、メディアデータにおける各バイナリーデータと 1 対 1 で対応させておき、例えば、メディアデータ内の埋め込みデータ 8 0 5 3 の区間に 1 を割り当て、正常パケットの区間には 0 を割り当てておく。この誤り通知データにより、アプリケーションデコーダは正確にパケットロスの発生を認識できる。この際、埋め込みデータ 8 0 5 3 は、必ずしもユニークワードである必要はない。但し、誤り通知データをアプリケーションデコーダに渡すための構成が必要となる。例えば、図 6 に示すように、パケットロス検出部 8 0 2 にて生成した誤り通知データパケット 7 0 6 を、誤り

通知出力部 8 0 4 からメモリ 2 0 7 に出力する仕組みとする。この場合、携帯電話端末の全体構成は図 1 7 のようになる。図 2 と図 1 7 の相違点は、誤り通知データパケット 7 0 6 がシステムデコーダ 2 0 5 からメモリ 2 0 7 に渡される点と、メモリ 2 0 7 に蓄積後、メディアデータと同期して、誤り通知データ 7 0 3 がアプリケーションデコーダ 2 0 9 に出力されている点である。図 4 3 ならびに図 4 4 は、それぞれ図 3 8 ならびに図 3 9 の携帯電話端末構成に誤り通知データの機能を加えたものを示す。RTP を使用する図 4 3 では、ペイロードタイプ別に、誤り通知データパケット 7 0 6 1 ならびに誤り通知データ 7 0 3 1 が処理されている。これに対し、TCP を使用しており、TCP パケットがペイロードタイプを持っていない図 4 4 では、アプリケーションデコーダ内のファイルデコーダにて誤り通知データが各メディアデコーダに分配される。なお、図 4 4 にて、ファイルフォーマットデータのヘッダ部にてパケットロスが発生している場合には、ファイルデコーダにてヘッダ部の情報を修正する必要がある。但し、より確実に再生処理を行うためには、図 6 あるいは図 4 1 の再送判定部 3 0 4 にて、ファイルヘッダ部でロスパケットが発生している場合にはデータが到達するまで再送要求をかけるか、ファイルの構成を送信側と端末側の間で決めておくといよい。

【 0 0 1 9 】

ここまでは、メディアデコーダにパケットロスを通知する方法について述べてきたが、この方法では、再生される画質ならびに音質は最終的にアプリケーションデコーダの性能に依存することになる。そのため、アプリケーションデコーダ自体が高度な誤り訂正手段を持っていない場合には、エラー検出用の埋め込みデータは逆にデコードの性能を落としてしまう可能性もある。また、アプリケーションデコーダが高度な誤り訂正手段を持つためには、誤り検出、誤り修復等の処理をアプリケーションデコーダに追加する必要がある、処理量も増える。そのため、メモリ要求、CPU パワーならびに消費電力等の要求が厳しい通信端末では、リアルタイム再生を実現する上で、簡易な誤り訂正手段しか採用できない場合も考えられる。そこで、本出願では、ファイル転送型のサービスに関して、エラー検出用埋め込みデータを利用してアプリケーションデコーダで処理される前に、パケットロスを伴うメディアデータをビデオ、オーディオ等の仕様や標準規格に

対応したデータに修正する方法についても提案する。

【0020】

図18に示す携帯電話端末2003は、図17の携帯電話端末2002に、メディアデータの修正処理構成を加えたものである。メモリ207へのメディアデータ入力以降の処理を以下に説明する。エラー・同期マーカ検出701では、メディア処理の標準規格にて定められている同期確保用の識別コード(同期マーカ)とシステムデコーダ205にて埋めこんだエラー検出用埋め込みデータ805を検索対象として、メモリ207内のメディアデータ208を順に検索する(図16で説明した誤り通知データを利用している場合には、誤り通知データ703も並列に検索し、メディアデータ内のエラー検出用埋め込みデータの替わりに、誤り通知データ内に記載されているエラー発生位置を示すデータ列を検出する)。そして、その何れかが検出された時点で、検出結果が同期マーカとエラー検出用埋め込みデータ(誤り通知データを利用している場合には、誤り通知データ内に記載されているエラー発生位置を示すデータ列)のいずれであったかを示す識別情報713をデータ修正処理部707に通知する。この際、同時にスイッチ705に、制御情報704を渡す。制御情報704は、検索開始位置から検出位置までのデータを第2メモリ710とデータ修正処理部707のいずれに出力すべきかを示す情報で、検出結果が同期マーカの場合には第2メモリ710に、検出結果がエラー検出用埋め込みデータ(誤り通知データを利用している場合には、誤り通知データ内に記載されているエラー発生位置を示すデータ列)の場合にはデータ処理部707にデータを出力することを示す。スイッチ705は、制御情報704の値に従ってスイッチを制御し、検出開始位置から次の同期マーカまでのメディアデータを第2のメモリ710あるいはデータ修正処理部707に渡るようにする(誤り通知データを利用している場合であり、かつ検出結果が誤り通知データ内に記載されているエラー発生位置を示すデータ列であるときには、メディアデータに対応する誤り通知データ703も同時にデータ修正処理707に渡される)。その後、メモリ207ではエラー・同期マーカ検出701にて検出したデータ位置までの情報を削除する(あるいは、検出した位置までポインタを移動する。誤り通知データも同様に処理)。データ修正処理部707では、入力デ

ータを調査し、規格方式に対応したデータに修正して第2メモリ710に出力する。この際、データ処理修正部707からスイッチ705に対して、スイッチ制御情報712が通知される場合がある(詳細は後述)。また、データ修正処理部707が、修正のために既に第2メモリ710に渡っているデータを必要とする場合には、処理済みデータ709を第2メモリから検索して入手する。第2メモリ710は、ユーザからの要求に応じてアプリケーションデコーダ209にメディアデータ711を出力する。この場合、アプリケーションデコーダ209は高度な誤り修正処理を必要としない(誤り通知データを使用しており、アプリケーションデコーダが誤り通知データを受信する仕組みを持っている場合には、誤り通知データをアプリケーションデコーダまで渡すことも可能である)。

【0021】

ここで説明したデータ修正処理は、システムデコーダにてメディアデータがすでにビデオデータとオーディオデータに分かれているような場合には、個々のメディアデータ毎に修正処理を行えばよいが、MP4ファイルのように複数のメディアデータが1個のファイルに統合されている場合には、ファイル内のデータ位置に応じて異なる修正処理を行う必要がある。具体的には、ファイル内の各メディアデータ部(ビデオデータ、オーディオデータ)ならびにファイルヘッダ部に、個々に修理処理を施す。そして、メディアデータ部の修正内容に従いファイルヘッダ部に記載されているデータ容量等を更書き換える必要がある。このように、ファイルヘッダをファイルフォーマットの仕様に対応したデータに修正することにより、ファイルフォーマットデコーダが各メディアデータを対応するデコーダに割り当てることができるようになる。

【0022】

次に、MPEG-4ビデオ規格を例にとって、データ修正処理部707での処理を詳細に説明する。説明に先立ち、まず、MPEG-4ビデオ規格の復号方式ならびにデータ構造について説明する。MPEG-4で扱う動画像の1フレームは、図23に示すように、1個の輝度信号(Y信号：2001)と2個の色差信号(Cr信号：2002, Cb信号：2003)にて構成されており、色差信号の画像サイズは縦横とも輝度信号の1/2となる。MPEG-4ビデオ規格では、動画像の各フレームを図23に示すような小ブロッ

クに分割し、マクロブロックと呼ばれるブロック単位で再生処理を行う。図24にマクロブロックの構造を示す。マクロブロックは16×16画素の1個のY信号ブロック2101と、それと空間的に一致する8×8画素のCr信号ブロック2102ならびにCb信号ブロック2103にて構成されている。なお、Y信号ブロックは、マクロブロックの復元過程では更に4個の8×8画素ブロック(21011, 21012, 21013, 21014)に分割して処理されることがある。MPEG-4ビデオの符号化アルゴリズムは、MC-DCT(動き補償-離散コサイン変換)と呼ばれており、上記に示したマクロブロック単位で再生処理がなされる。動き補償とは、前フレームから対象マクロブロックの内容と似通った部分を抜き出し、その動き量を符号化する方法のことであり、動き補償により抜き出された前フレームのブロック領域と原画像の符号化対象ブロック(21011, 21012, 21013, 21014, 2102, 2103)との差分ブロック画像が、DCTにて周波数変換された後、各変換係数が量子化して符号化される。さらに細かく見るとMPEG-4ビデオ規格にはフレーム内符号化(イントラ符号化)、フレーム間符号化(予測符号化)、双方向符号化と呼ばれる符号化方法がある。イントラ符号化とは、動き補償後の差分ブロック画像ではなく、符号化対象の入力ブロック画像に対して直接DCTを施し、各変換係数を量子化・符号化するデータ圧縮方法であり、全てのマクロブロックに対してイントラ符号化を適用したフレームをI-VOP(Intra-coded Video Object Plane、矩形画像ではVOPはフレームと同義)と呼ぶ。I-VOPは、過去のフレームの復号情報を必要としないため、ランダムアクセス時の復号開始フレームとして使用される。予測符号化とは、MC-DCTを利用した圧縮方法で、特に時間的に過去フレームに対してMCを行う符号化方法のこと。そして、フレーム内のマクロブロックに対して、予測符号化あるいはイントラ符号化を施したフレームをP-VOP(Predictive-coded VOP)と呼ぶ。これ以外に、時間的に過去と未来のフレームの情報を用いてMCを行う方法(双方向符号化)もあり、この符号化方法を使用したフレームはB-VOP(Bidirectionally predicted-coded VOP)と呼ばれている。

【0023】

図19は一般的なMPEG-4ビデオ復号化のブロック図である。入力されたメディ

アデータは、まず、符号解読部 5 0 1 によって、バイナリーコードから意味のある復号情報に変換される。そして、DCT係数の量子化データに関連する情報は、逆量子化部 5 0 2 に送られ、DCT係数データに復号された後、逆DCT器 5 0 3 にて差分マクロブロック画像に復元される。一方、動き量に関する情報は動き補償部 5 0 4 に送られる。動き補償器 5 0 4 では、前フレームのデータが蓄積されているフレームメモリ 5 0 7 から、復号された動き量に従って予測マクロブロック画像が復元される。このようにして復元された差分マクロブロック画像と予測マクロブロック画像は、加算器 5 0 5 にて復元マクロブロック画像が生成され、合成部 5 0 6 にて復元フレーム画像に合成される。復元されたフレーム画像は、表示処理部に出力されると共に、次フレームの動き補償のために、フレームメモリ 5 0 7 に保存される。

図 2 0 は、誤り検出ならびに誤り修正処理機能を持つMPEG-4ビデオ復号化のブロック図例である。符号解読部 5 0 1 2 では、入力データの解析と共に誤り検出が行われる。誤り検出にはいろいろな方法があるが、一般的には、MPEG-4ビデオ符号化のコードブックに存在しない可変長コードの検出によって行われる。データ誤りが検出された場合には、符号解読部 5 0 1 2 は対象データを修復処理部 5 0 9 に出力する。また、符号解読部 5 0 1 2 は、データ誤りを検出した時点で予測マクロブロック画像の出力制御情報 5 1 0 をスイッチ 5 0 8 に通知し、予測マクロブロック画像が修復処理部 5 0 9 に渡るように制御する。符号解読部 5 0 1 2 における誤り検出処理ならびに修復処理部 5 0 9 の処理内容は標準化されていないため、伝送エラーが含まれるデータの再生画質は、各製品の仕様に依存することになる。

【 0 0 2 4 】

図 2 1 , 図 2 2 ならびに図 2 5 は、MPEG-4ビデオ符号化の基本的なデータ構造を示したものである。図 2 1 が全体のデータ構造、図 2 2 がフレームヘッダのデータ構造、図 2 5 が各マクロブロックのデータ構造を示している。図 2 1 において、VOSヘッダはMPEG-4ビデオ製品の適用範囲を決めるプロファイル・レベル情報、VOヘッダはMPEG-4ビデオ符号化のデータ構造を決めるバージョン情報、VOLヘッダは画像サイズ、符号化ビットレート、フレームメモリサイズ、適用ツール

等の情報を含んでいる。いずれも受信した符号化データを復号する上で必須の情報である。GOVヘッダには時刻情報が含まれているが、これは必須の情報ではなく省略も可能である。VOSヘッダ、VOヘッダ、VOLヘッダ、GOVヘッダとも32ビットのユニークワードから始まるため、容易に検索できる。シーケンスの終了を示すEnd code of VOSも32ビットのユニークワードである。これらのユニークワードは、23個の0と1個の1で始まり、その24ビットに続く2バイトのデータがその区切りの種類を示すような構造となっている。VOPには、各フレーム(MPEG-4ビデオではVOPと呼ぶ)のデータが含まれている。VOPは、図22に示したVOPヘッダから始まり、図25に示される構造のマクロブロックデータが、フレームの上から下、左から右の順で続く構造となっている。図22がVOPヘッダのデータ構造である。VOPスタートコードと呼ばれる32ビットのユニークワードから始まる。vop_coding_typeがVOPの符号化タイプ(I-VOP, P-VOP, B-VOP等)を表し、それに続くmodulo_time_baseならびにvop_time_incrementがそのVOPの出力時刻を示すタイムスタンプとなっている。modulo_time_baseは秒単位の情報、vop_time_incrementは秒以下の情報である。vop_time_incrementの精度に関する情報はVOLヘッダのvop_time_increment_resolutionに示されている。modulo_time_base情報は、前VOPの秒単位の値と現VOPの秒単位の値との変化分を示す値で、その変化分の数だけ1が符号化される。つまり、秒単位の時刻が前のVOPと同じ場合には0が、1秒分異なる場合には10、2秒分異なる場合には110が符号化されている。vop_time_increment情報は、各VOPにおける秒以下の情報をvop_time_increment_resolutionが示す精度で示している。intra_dc_vlc_thrには、イントラ符号化マクロブロックにおけるDCT係数のDC成分が、AC成分とは異なる符号化テーブルで符号化されているか、同じ符号化テーブルで符号化されているかを識別するための情報が含まれている。このintra_dc_vlc_thrの値と各マクロブロックにおけるDCT係数の量子化精度から、いずれの符号化テーブルを用いるかがマクロブロック単位で選択される。vop_quantは、DCT係数を量子化する際の量子化精度を示す値であり、そのフレームの量子化精度初期値となる。vop_fcode_forwardとvop_fcode_backwardはMCにおける動き量の最大範囲を示している。図25がマクロブロックの基本データ構造(I-VOPとP-VOP)である。not_codedはP-VOPのみ

で使用される 1 ビットのフラグで、以降にそのマクロブロックに関するデータが続いているかどうかを示している、0 の場合には、そのマクロブロックに関するデータが続くことを意味し、1 の場合には、以降に続くデータが次のマクロブロックに関するデータであり、そのマクロブロックの復号信号は前フレームの同じ位置からコピーすることにより生成することを示している。mcbpc は 1～9 ビットの可変長符号であり、そのマクロブロックの符号化タイプを示す mb_type と、2 個の各色差ブロック内に符号化すべき量子化 DCT 係数 (0 以外の値) があるか否かを示す cbpc (イントラ符号化ブロックの場合には、量子化 DCT 係数の AC 成分があるか否かを示す) を 1 個の符号で表している。mb_type が示す符号化タイプには、intra, intra+q, inter, inter+q, inter4v (inter4v は、輝度信号の動き補償を行う単位が図 2 4 の 2 1 0 1 ではなく、2 1 0 1 1 から 2 1 0 1 4 の 4 個の小ブロック単位であることを示す), stuffing の 5 種類があり、intra と intra+q がイントラ符号化、inter、inter+q ならびに inter4v が予測符号化、stuffing は符号化レートを調整するためのダミーデータであることを示している。+ q は、DCT 係数の量子化精度が前ブロックの値(quant)あるいは初期値(vop_quant、フレームの最初の符号化マクロブロックに適用)から変更されることを示している。なお、stuffing の場合には、図 2 5 内の ac_pred_flag 以降のデータは省略されており、復号された mcbpc と not_coded の値も再生画像の合成には反映されない。ac_pred_flag は、mb_type がイントラ符号化であった場合のみ含まれる情報で、DCT 係数の AC 成分に対して、周囲のブロックからの予測を行うか否かを示す。この値が 1 の場合は、AC 成分の量子化再生値の一部が周囲のブロックからの差分値となっている。cbpy は 1～6 ビットの可変長符号で、4 個の各輝度ブロック内に符号化された量子化 DCT 係数 (0 以外の値) があるか否かを示している (cbpc と同様に、イントラ符号化ブロックの場合には、量子化 DCT 係数の AC 成分があるか否かを示す)。dquant は、mb_type が intra+q あるいは inter+q であった場合にのみ存在し、前ブロックの量子化精度の値からの差分値を示しており、quant+dquant がそのマクロブロックの quant となる。動きベクトルの符号化に関する情報は、mb_type が予測符号化の場合のみ含まれる。Intra 差分 DC 成分は、mb_type がイントラ符号化であり、かつ use_intra_dc_vlc が 1 の場合にのみ含まれる情報である。MPEG-4 ビ

デオでは、イントラ符号化ブロックにおけるDCT係数のDC成分は、周囲のマクロブロックにおけるDCT係数のDC成分との差分値が量子化されている。量子化の方法もAC成分とは異なり、符号化方法もAC成分とは別の方法が用意されている。但し、`use_intra_dc_vlc`を0とすることによりDC成分の量子化値もAC成分の量子化値と同様の符号化方法を適用することが可能となる。なお、`use_intra_dc_vlc`の値は、VOPヘッダにて定義される`intra_dc_vlc_thr`と当該マクロブロックの`quant`にて決定される。Intra AC成分 or Inter DC&AC成分に関しては、「`cbpy`, `cbpc`にて、DCTの量子化係数に0以外の値が存在することが示されている」ブロックのみ情報を持つ。

【 0 0 2 5 】

上記で示した図25のマクロブロックデータ構造では、伝送誤り等の影響で1度データの同期が失われると、次フレームのスタートコードまでデータ同期を回復することができない。そこで、MPEG-4ビデオ規格では、誤り耐性用のデータ構造を別途用意している。具体的には、ビデオパケットならびにデータ分割と呼ばれる処理が用意されており、VOLヘッダにてその適用が指定される。ビデオパケットとは、数個のマクロブロックを集めたデータ単位のこと、各ビデオパケットデータの先頭には同期マーカとフレーム内での位置情報を含むビデオパケットヘッダが配置されている。また、ブロック間でのデータの予測もパケット内で閉じている。そのため、データ同期が失われた場合でも、その影響を1個のパケット内にとどめることが可能で、次のビデオパケットの先頭で同期を回復することも可能である。図26がビデオパケット分割の例である。図26では、2301, 2302, 2303, 2304, 2305の5個のビデオパケットに分割されている。フレームの第1パケット2301を除く各パケットデータの先頭には、図28に示されるビデオパケットヘッダが配置され、フレームの第1パケットについては、図22のVOPヘッダが配置される。パケットヘッダのデータ構造は、`resync_marker`と呼ばれる同期マーカから始まる。この同期マーカは、17~23ビットのユニークワードで、そのデータサイズはフレームの符号化タイプや動き補償時の動き量の範囲から一意に決まるものである。`macroblock_number`はパケット内の先頭マクロブロックのフレーム内位置を示すものである。この最初の2

個のデータにより、同期は完全に回復する。quant_scaleは、そのビデオパケットにおけるDCT係数の量子化精度の初期値である。この値が当該ビデオパケットの最初の符号化マクロブロックに対する前マクロブロックのquantとして適用される。header_extension_codeは、modulo_time_baseからvop_fcode_backwardまでのデータをビデオパケットヘッダに含めるか否かを示すための1ビットのフラグで、1の場合にはmodulo_time_base以降が含まれる。なお、modulo_time_base以降のデータについては、VOPヘッダと同じ値にすることが規定されており、VOPヘッダ内の情報が伝送エラーの影響を受けているかどうかを調査する等の役割を持つ。これに対してデータ分割とは、図25にて説明したマクロブロックのデータ構造にて示される符号の位置をビデオパケット内で入れ替えることにより、各マクロブロックの重要データが伝送エラーにて解読不能になる確率を減らす機能のことである。

【0026】

図27は、フレームタイプがI-VOPの場合について、ビデオパケット内のデータ分割構造を示したものである。図27におけるビデオパケットヘッダ2401のデータ構造は、図28に示したとおりである。図29は、図27の優先データ部2402に配置されるデータについて、1マクロブロック分の優先データ部のデータ構造24021を示したものである。この図29のデータは、図25に示したマクロブロックデータから再生ブロックを合成する上で重要なデータのみを抜き出したものと言える。図27の優先データ部2402には、ビデオパケット内の各マクロブロックについて図29に示したデータ)のみを抜き出し、順次配置される。但し、dquantとIntra差分DC成分については、図中に示した条件が成立しているマクロブロックのみに存在する。図27のdc_marker2403は19ビットのユニークワードである。図30は、図27のAC成分制御情報2404に配置されるデータについて、1マクロブロック分のAC成分制御情報部のデータ構造24041を示したものである。図31は、図27のAC成分情報2405に配置されるデータについて、1マクロブロック分のAC成分情報部のデータ構造24051を示したものである。いずれも、ビデオパケット内の各マクロブロックについて、図25に示したマクロブロックデータから図29または図30に示したデー

タのみを抜き出し、順次配置される。但し、各マクロブロックにおけるAC成分情報の存在はcbpy, cbpcの値に依存して変化する。このような構造を持つことにより、2404あるいは2405にて伝送誤りが発生しても、dc_markerの復号にて、優先データ部2402が同期崩れなく復号されていることが判定できるようになる。これにより、優先データ部のデータのみを用いて、当該ビデオパケット内の全マクロブロックについて、大まかな特徴を持つブロック画像を合成することが可能となる。また、優先データ部2402にて伝送誤りが発生しても、dc_markerを検索することにより、ビデオパケット内で同期を回復させることも可能である。但し、優先データ部のデータが壊れているとAC成分制御情報部2404とAC成分情報部2405に伝送エラーが無い場合であっても、mb_type, cbpcならびにdquant等の情報が使えないため、必ずしも正確に復号できるとは限らない。

【0027】

図32は、フレームタイプがP-VOPの場合について、ビデオパケット内のデータ分割構造を示したものである。図32におけるビデオパケットヘッダ2401のデータ構造は、図28に示したとおりである。図33は、図32の優先データ部2902に配置されるデータについて、1マクロブロック分の優先データ部のデータ構造29021を示したものである。この図33のデータは、図25に示したマクロブロックデータから再生ブロックを合成する上で重要なデータのみを抜き出したものと言える。図32の優先データ部2902には、ビデオパケット内の各マクロブロックについて、図33に示したデータのみを抜き出し、順次配置される。但し、not_codedの値が1のマクロブロックについては、mcbpcと動きベクトル情報は省略される。また、mb_typeがイントラ符号化あるいはstuffingのマクロブロックについては、動きベクトル情報は省略される。図32のmotion_marker2903は17ビットのユニークワードである。図34は、図32のIntra差分DC成分情報&AC成分制御情報2904に配置されるデータについて、1マクロブロック分のIntra差分DC成分情報&AC成分制御情報のデータ構造29041を示したものである。図35は、図32のIntra AC成分 or Inter DC&AC成分情報2905に配置されるデータについて、1マクロブロック分のIntra AC成分

r Inter DC&AC成分情報のデータ構造 2 9 0 5 1 を示したものである。いずれも、ビデオパケット内の各マクロブロックについて、図 2 5 に示したマクロブロックデータから図 3 4 または図 3 5 に示したデータのみを抜き出し、順次配置される。但し、not_coded が 1 のマクロブロックについては、マクロブロック内に図 3 4 ならびに図 3 5 に示すデータが存在しないため、省略される。また、各マクロブロックにおける Intra 差分 DC 成分情報 & AC 成分制御情報 2 9 0 4 1 内の ac_pred_flag, dquant ならびに Intra 差分 DC 成分については、図中に示した条件が成立しているマクロブロックのみに存在する。さらに、各マクロブロックにおける Intra AC 成分 (mb_type がイントラ符号化の場合) or Inter DC&AC 成分情報 (mb_type が予測符号化の場合) の存在は cbpy, cbpc に依存して変化する。

【 0 0 2 8 】

図 3 6 に、データ修正処理部 7 0 7 での処理をビデオパケットならびにデータ分割の機能を用いたデータを例として示す。最初に、図 1 8 のエラー・同期マーカ検出部 7 0 1 から、データ検索の結果が同期マーカ (各種スタートコードあるいは resync_marker) あるいはエラー検出用埋め込みデータのいずれであったかを示す識別情報 7 1 3 を受け、検出結果が同期マーカであったか否かを判定する (ステップ 3 3 0 1)。検出結果が同期マーカであった場合には、検索開始位置から検出位置までのデータ、すなわち 1 ビデオパケット分のデータが、図 1 8 のエラー・同期マーカ検出部 7 0 1 からスイッチ 7 0 5 に通知される制御情報 7 0 4 により第 2 メモリ 7 1 0 に出力される。データ処理部 7 0 7 では、第 2 メモリ 7 1 0 へ入力されたデータを保存することを示す情報を通知し、処理を終了する (ステップ 3 3 1 3)。一方、検出結果がエラー検出用埋め込みデータであった場合には、図 1 8 のエラー・同期マーカ検出部 7 0 1 からスイッチ 7 0 5 に通知される制御情報 7 0 4 により、検索開始位置から次の同期マーカまでのメディアデータ 2 0 8 がメモリ 2 0 7 からデータ修正処理部 7 0 7 に得られる (ステップ 3 3 0 2)。次に、データ処理部 7 0 7 では、獲得したメディアデータ 2 0 8 を解析し、エラー区間をチェックする。チェックは、エラー区間が VOL ヘッダを含むか? (ステップ 3 3 0 3)、ステップ 3 3 0 3 で VOL ヘッダを含んでいない場合、エラー区間が VOP ヘッダを含むか? (ステップ 3 3 0 4)、ステップ 3 3 0 4

でVOPヘッダを含んでいない場合、エラー区間がビデオパケットヘッダ内のマクロブロック位置データ(図28のmacroblock_number)を含むか?(ステップ3305)、ステップ3303でVOLヘッダを含んでいた場合、そのVOLヘッダは再送ヘッダであるか?(ステップ3306)、ステップ3304でVOPヘッダを含んでいた場合、vop_coding_typeがB-VOPであるか?(ステップ3307)の5つで行う。ここでステップ3307については、VOPヘッダ内のvop_coding_type情報がエラー区間に含まれていた場合、当該フレーム内のパケットヘッダに含まれるvop_coding_type情報(図28のvop_coding_type)をこの判定に利用してもよい。

エラー区間チェックの結果、ステップ3306にて、エラー区間が再送VOLヘッダを含むと判定される場合には、VOSヘッダ、VOヘッダならびにVOLヘッダの組み合わせを捨てることを示すエラー修正情報708を図18の第2のメモリ710に通知し、処理を終了する(ステップ3315)。この際、VOSヘッダ、VOヘッダが既に第2のメモリ710に移動している場合には、第2のメモリはエラー修正情報708に応じて、対応するヘッダ情報を削除する。エラー区間チェックの結果、4)にて、エラー区間がシーン先頭のVOLヘッダを含むと判定される場合には、エラーを含まない次のVOLヘッダまでのメディアデータがデータ修正処理部707に入力されるようにスイッチ制御情報712を制御し、図18のスイッチ705に通知する(ステップ3314)。そして、次のVOLヘッダまでの情報を削除し、VOLヘッダ情報のみをエラー修正情報708として第2のメモリに渡す。

また、VOLヘッダ後のVOPがI-VOPでない場合には、次のI-VOPまでのメディアデータがデータ修正処理部707に入力されるようにスイッチ制御情報712を制御し、スイッチ705に通知する。そして、シーン先頭のVOLヘッダに続くVOPの種類がI-VOPとなるようにデータの削除を行う。エラー区間チェックの結果、ステップ3307にて、vop_coding_typeがB-VOPであると判定できる場合には、次のVOPヘッダまでのメディアデータがデータ修正処理部707に入力されるように、スイッチ制御情報712を図18のスイッチ705に通知し、次のスタートコードまでのデータを削除する(ステップ3315)。エラー区間チェックの結果、5)にて、vop_coding_typeがB-VOPでないと判断できる場合には、以下の(1)から(4)の処理をおこなう(ステップ3317)。

- (1) 次のヘッダ情報付き同期マーカあるいはスタートコードまでのデータを捨てるようにスイッチ制御情報 7 1 2 を制御し、スイッチ 7 0 5 に通知する。
- (2) 現マクロブロックの位置(macroblock_number)を 0 とする。
- (3) ヘッダ情報付き同期マーカがフレーム内に含まれる場合、ヘッダ情報から VOP ヘッダを生成する。
- (4) ヘッダ情報付き同期マーカがフレーム内に含まれていない場合、vop_coded を 0 として VOP ヘッダを作成する。次フレームの vop_time_increment の値が、前フレームよりも小さい場合には、modulo_time_base を 1 0 とし、vop_time_increment を 0 とする。それ以外の場合には、modulo_time_base を 0 とし、vop_time_increment を前フレームの値に 1 を加えた値とする。

なお、処理(3)、(4)に示すヘッダ情報付き同期マーカの検索の際には、メモリ 2 0 7 内のメディアデータ 2 0 8 を利用する。処理(4)を実行した場合には、修正データの終端部にバイトアライン処理を施し(ステップ 3 3 1 1)、第 2 のメモリ 7 1 0 にバイトアライン化した修正データ 7 0 8 を出力して、処理を終了する(ステップ 3 3 1 2)。エラー区間チェックの結果、ステップ 3 3 0 5 にて、エラー区間がビデオパケットヘッダ内のマクロブロック位置データを含んでいた場合、第 2 のメモリから前ビデオパケット(VP)の処理済みデータ 7 0 9 を読み込み、前 VP ヘッダ内の macroblock_number と前 VP の符号化 MB 数を調査する(ステップ 3 3 1 6)。この処理により、当該 VP ヘッダの macroblock_number を再現することが可能となる。

【 0 0 2 9 】

エラー区間処理にて、ステップ 3 3 0 3 とステップ 3 3 0 4 の処理を満たさない場合と、ステップ 3 3 0 7 にて VOP 符号化タイプが B-VOP ではないと判定でき、かつステップ 3 3 1 7 の処理にて処理(3)を実行した場合については、マクロブロックデータの修正処理を行う。まず、ステップ 3 3 0 3 とステップ 3 3 0 4 の処理を満たさない場合については、当該ビデオパケットの第 1 MB の位置(macroblock_number)を検出する(ステップ 3 3 0 8)。ステップ 3 3 0 4 を満たす場合に関しては、ステップ 3 3 1 7 の処理(2)において、当該ビデオパケットの第 1 MB の位置を 0 と設定しているため、ステップ 3 3 0 8 は省略できる。次に、メモリ

207内のメディアデータ208を検索し、次VPの第1MBの位置、すなわち当該VP内における処理MB数を検出する（ステップ3309）。なお、検索の結果、次VPのresync_markerの前にスタートコードが検出された場合には、当該VPの最終MBが当該VOPの最終MBであると判定する。そして、ビデオパケット内の符号化マクロブロック数情報とビデオパケット内のメディアデータ208を用いて、図37に示すマクロブロックデータ修正処理3310を実行する。最後に修正データの終端部にバイトアライン処理3311を施し、第2のメモリ710にバイトアライン化した修正データ708を出力して、処理を終了する（ステップ3312）。

【0030】

次に、図37のマクロブロックデータ修正処理3310について説明する。パケット内の各マクロブロックのデータ修正方法として、本明細書では以下の3つの方法を用いることにする。

処理I：not_codedフラグを1とする。MB内の元データは削除。

処理II：MB内の差分DC成分を全て0とし、mb_typeを「intra」、cpby, cbpcを符号化ブロックなしに設定する。MB内の元データは削除。

処理III：cpby, cbpcを符号化ブロックなしに設定する。さらに、I-VOPの場合には、ac_pred_flagを0とし、AC成分データは削除する。P-VOPの場合には、mb_typeがイントラ符号化(intra, intra+q)のときは処理I、mb_typeが予測符号化(inter, inter+q, inter4v)のときは、Inter DC&AC成分データを削除する。

処理方法の選択手順は、図27あるいは図32のどの部分でエラーが発生しているかによって決める。そこで、入力されたビデオパケット内のメディアデータ208をチェックし、エラー区間を、エラー区間がビデオパケットヘッダ(図27の2401あるいは図32の2401)を含むか？（ステップ3401）、ステップ3401でビデオパケットヘッダを含んでいない場合、エラー区間が優先データ部(図27の2402あるいは図32の2902)を含むか？（ステップ

3403)、ステップ3403でビデオパケットヘッダを含んでいる場合、エラー区間が中間マーカ(図27の2403あるいは図32の2903)を含んでいるか?(ステップ3404)のように3つのステップでチェックする。

【0031】

ステップ3401にて、エラー区間がビデオパケットヘッダを含むと判断される場合には、ステップ3402処理を実行し、処理を終了する。具体的には、まず、図36のステップ3308の処理で検出した第1MBの位置に従って、ビデオパケットヘッダ部を修正する。その際、quant_scaleがエラー区間に含まれていた場合には、前VPパケットと同じ値に定める。次に、ビデオパケット内の各MBについてデータを生成する。生成方法としては、VOPの符号化タイプがI-VOPの場合には処理II、P-VOPの場合には処理Iの方法を利用する。さらに、優先データ部の後に中間マーカ(I-VOPの場合にはdc_marker、P-VOPの場合にはmotion_marker)を追加する。なお、このケースでは、パケット内の元のデータは全て廃棄し、修正データのみ出力する。

【0032】

ステップ3403にて、エラー区間が優先データを含まないと判断される場合には、ステップ3407の処理を実行し、処理を終了する。具体的には、DCTデータ部(図27の2404と2405、あるいは図32の2904と2905)が正常なMBについては、優先データ部ならびにDCTデータ部のデータをそのまま保存する。DCTデータ部がエラー区間に含まれているMBについては、処理IIIの方法でデータを修正する。なお、次のVPヘッダのresync_markerがエラー区間に含まれている場合には、図36のステップ3309の処理にて検出したVPは、実際には2つ後のVPとなり、次VP内のMBも当該VP内のMBとして扱う必要が生じる。このように優先データ部の情報がないMBについては、VOPの符号化タイプがI-VOPなら処理II、P-VOPなら処理Iを用いて優先データ部ならびにAC成分制御情報部(I-VOPの場合のみ)のMBデータを生成し、優先データ部の直後に中間マーカを挿入する。

【0033】

ステップ3404にて、エラー区間が中間マーカを含んでいると判断される場

合には、ステップ 3 4 0 5 の処理を実行し、処理を終了する。具体的には、優先データ部が正常なMBについては

処理 I I I、優先データ部がエラー区間に含まれているI-VOP上のMBについては処理 I I、優先データ部がエラー区間に含まれているP-VOP上のMBについては処理 I の方法でデータを修正する。さらに、優先データ部の直後に中間マーカ(I-VOPの場合にはdc_marker、P-VOPの場合にはmotion_marker)を挿入する。

【 0 0 3 4 】

ステップ 3 4 0 4 にて、エラー区間が中間マーカを含まないと判断される場合には、ステップ 3 4 0 6 の処理を実行し、処理を終了する。具体的には、優先データ部、DCTデータ部共に正常なMBについては、優先データ部ならびにDCTデータ部のデータをそのまま保存する。優先データ部のみエラー区間に含まれているMBについては、VOPの符号化タイプがI-VOPなら処理 I I、P-VOPなら処理 I を用いて優先データ部ならびにAC成分制御情報部(I-VOPの場合のみ)のMBデータを修正する。DCTデータ部のみがエラー区間に含まれているMBについては、処理 I I Iの方法でデータを修正する。なお、次のVPヘッダのresync_markerまでがエラー区間に含まれている場合には、図 3 6 のステップ 3 3 0 9 の処理にて検出したVPは、実際には2つ後のVPとなり、次VP内のMBも当該VP内のMBとして扱う必要が生じる。このような優先データ部情報がないMBについては、VOPの符号化タイプがI-VOPなら処理 I I、P-VOPなら処理 I を用いて優先データ部ならびにAC成分制御情報部(I-VOPの場合のみ)のMBデータを生成する。

【 0 0 3 5 】

ここまで述べてきたように、データのエラー修正機能はスタートコード、同期マーカならびに中間マーカを起点として行われる。従って、これらのユニークワードを区切りとし、例えば、中間マーカとDCTデータ部が1個のパケットに含まれないように、送信側で伝送パケットを構成しておくことで、本発明にて述べている方法はより有効に機能する。

本発明には次のような場合も含まれる。

1 つめは、本発明のエラー検出用埋め込みデータの埋め込み方法は、パケットロスに伴う通信手段で利用でき、かつ各パケット内のヘッダ部にそのパケットのシ

ーケンス番号(パケットを合成する順番)が含まれているプロトコルであれば、送信端末とのコネクションレス型、コネクション型に関わらず適用可能である。よって、本発明のエラー検出用埋め込みデータの埋め込み方法は、本明細書で示したTCPとUDP/RTPに特化するものではない。また、回線交換方式であっても、決まった区切りで伝送エラーを修正するため、アプリケーションデコーダへの出力にエラー修正結果を反映させることが可能である。このように、受信データの特定部分に通信エラーが発生していると判定できる場合には本発明のエラー検出用データ埋め込み方法は適用できる。

2つめは、本発明のエラー検出用埋め込みデータの埋め込み方法は、パケットロス部等の影響で発生するデータ誤りの位置を、明示的にアプリケーションデコーダに伝えるための識別コードをメディアデータ内に埋め込むことに特徴があり、その具体的な数値は図1、図15ならびに図16に示した数値には依存しない。パケットロス部分のデータを再生するアプリケーションデコーダがエラーと判定できる埋め込みデータの発生手段ならびに埋め込み構成は本発明に含まれる。また、埋め込み位置の検出方法ならびに構成も図2と図6に限定されるものではない。

3つめは、本発明のメディアデータ修正方法は、パケットロス等の伝送誤りの影響を受けた画像、音声等のメディアデータを、アプリケーションデコーダに入力する前に、そのメディアデータを解読し、再現するための仕様・規格に従ったデータに修正することに特徴がある。従って、本明細書では、MPEG-4ビデオ規格を例にとって説明したが、このようなデータ修正は、他の動画像符号化方式でも同様のことは実現可能であり、様々な静止画符号化方式、音声符号化方式ならびにデータ符号化方式、さらにMP4のようなメディア統合ファイルフォーマットに対しても実現可能である。よって、アプリケーションデコーダ(ビデオデコーダ、オーディオデコーダ、シーン記述デコーダ、ファイルフォーマットデコーダならびにこれらの組み合わせ等)に入力する前に、データを仕様準拠・規格準拠に変換する方法は全て本発明に含まれる。また、本発明を実現するための構成は図18に限定されるものではない。例えば、メモリを1個とし、随時データを修正して入れ替える方法やエラー検出機能をデータ修正処理部に含めるような場合も本

発明に含まれる。

4 つめは、本明細書では、携帯電話端末を例にとって説明したが、本発明はその他の無線端末や配信サーバ等でも実現できる。例えば、無線通信から有線通信へデータ変換部・変換機に本発明のデータ修正方法を適用することは有効である。

【 0 0 3 6 】

【発明の効果】

本発明により、アプリケーションデコーダにてパケットロスの影響を受けているデータ位置を正確に検出することが可能となる。また、アプリケーションデコーダに入力する前にメディアデータをデコーダの仕様・規格に修正することで、アプリケーションデコーダにおける誤り修正処理の負担を減らし、データの再生速度を高速化することが可能となる。

【図面の簡単な説明】

【図 1】

本発明のユニークワードを埋め込んだシステムデコーダの出力データ。

【図 2】

携帯電話端末の内部構成を示す図。

【図 3】

送信パケットの一例図。

【図 4】

パケットロスが発生していない場合のシステムデコーダからの出力メディアデータの一例図。

【図 5】

パケットロスが発生した場合の、システムデコーダからの出力メディアデータの一例図。

【図 6】

エラー検出用埋め込みデータの埋め込み処理を実現するためのシステムデコーダ。

【図 7】

本発明のシステムデコーダにおける、パケット処理部からメディアデータ再構

成部への出力データ。

【図 8】

本発明のシステムデコーダにおいて、パケット処理部の処理手順の一例。

【図 9】

IPv4パケットのデータ構成。

【図 1 0】

PPPフレームのデータ構成。

【図 1 1】

RLPフレームのデータ構成。

【図 1 2】

TCPパケットのデータ構成。

【図 1 3】

UDPパケットのデータ構成。

【図 1 4】

RTPパケットのデータ構成。

【図 1 5】

本発明のエラー検出用埋め込みデータの一例。

【図 1 6】

本発明の誤り通知データを付加するエラー検出用埋め込みデータの一例。

【図 1 7】

誤り通知データの出力機能を持つ携帯電話端末の内部構成図。

【図 1 8】

本発明のデータ修正処理を実現する携帯電話端末の内部構成図。

【図 1 9】

MPEG-4デコーダのブロック図の一例。

【図 2 0】

誤り訂正機能を持つMPEG-4デコーダのブロック図の一例。

【図 2 1】

MPEG-4ビデオ符号化ビットストリームの一列。

【図 2 2】

MPEG-4ビデオ符号化ビットストリームにおけるVOPヘッダのデータ構成。

【図 2 3】

MPEG-4ビデオ符号化におけるマクロブロック分割を示す図。

【図 2 4】

MPEG-4ビデオ符号化におけるマクロブロックの構成。

【図 2 5】

MPEG-4ビデオ符号化ビットストリームにおけるMBデータのデータ構成。

【図 2 6】

MPEG-4ビデオ符号化におけるビデオパケット分割例。

【図 2 7】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケットデータ(I-VOP)のデータ構造。

【図 2 8】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケットヘッダのデータ構造。

【図 2 9】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケット優先データ部(I-VOP)のデータ構造。

【図 3 0】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケットAC成分制御情報部のデータ構造。

【図 3 1】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケットAC成分情報部のデータ構造。

【図 3 2】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケットデータ(P-VOP)のデータ構造。

【図 3 3】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケット優先データ部(P-VOP)のデータ構造。

【図 3 4】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケットIntra差分DC成分情報&AC成分制御情報部のデータ構造。

【図 3 5】

MPEG-4ビデオ符号化ビットストリームにおけるビデオパケットIntra AC成分 or Inter DC&AC制御情報部のデータ構造。

【図 3 6】

本発明のメディアデータ修正アルゴリズムを示すフローチャート。

【図 3 7】

本発明のマクロブロックデータ修正アルゴリズム示すフローチャート。

【図 3 8】

RTP使用時における携帯電話端末の内部構成を示す図。

【図 3 9】

TCP使用時における携帯電話端末の内部構成を示す図。

【図 4 0】

アプリケーションデコーダでの処理状況を考慮して、不正パケットの再送判定を行うことが可能な携帯電話端末の内部構成を示す図。

【図 4 1】

エラー検出用埋め込みデータの埋め込み処理を行うシステムデコーダ。

【図 4 2】

本発明のシステムデコーダにおいて、擬似ストリーミングにおけるパケット処理部の処理手順の一例。

【図 4 3】

RTP使用時における誤り通知データの出力機能を持つ携帯電話端末の内部構成図。

【図 4 4】

TCP使用時における誤り通知データの出力機能を持つ携帯電話端末の内部構成

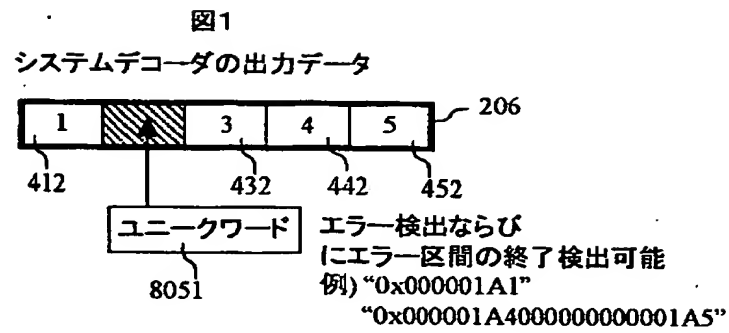
図。

【符号の説明】

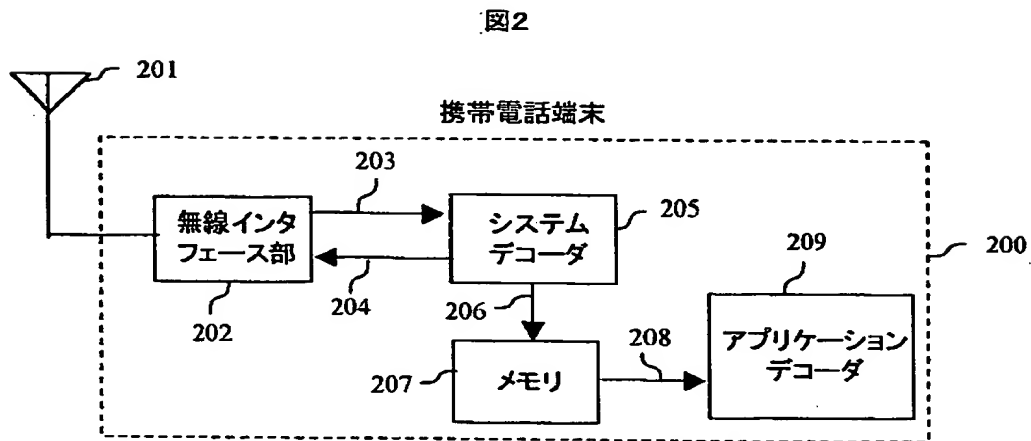
205…システムデコーダ、207、710…メモリ、209…アプリケーションデコーダ、301…パケット処理部、303…再送判定結果・シーケンス番号、304…再送判定部、305…応答確認部、307…ペイロードデータ、308…メディアデータ再構成部、801…タイムスタンプ・シーケンス番号・ペイロードタイプ、802…パケットロス検出部、701…エラー・同期マーカ検出部、805、8051、8052、8053…エラー検出用埋め込みデータ、806、807…パケットロスが発生したシーケンス番号ならびにペイロードタイプ、807…パケットロスが発生したシーケンス番号、706…誤り通知データ、705…スイッチ、707…データ修正処理部、3310…マクロブロックデータ修正部、3311…バイトアライン処理部

【書類名】 図面

【図 1】

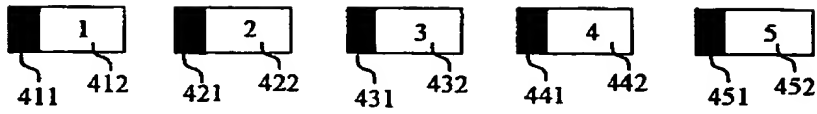


【図 2】



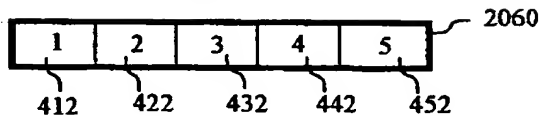
【図 3】

図3



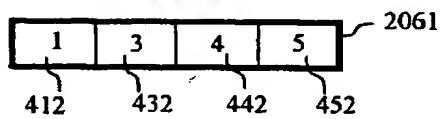
【図 4】

図4

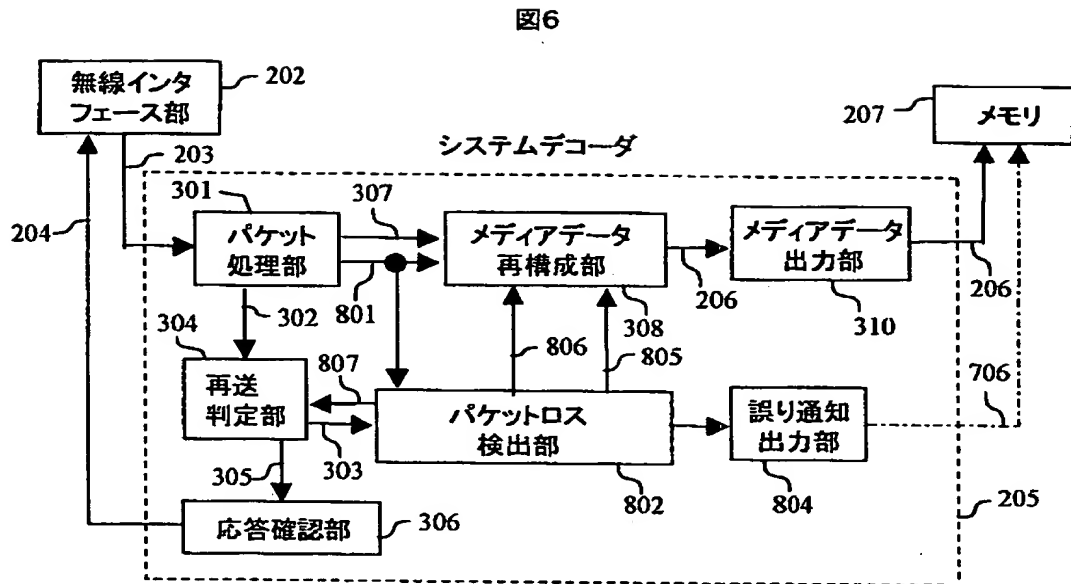


【図 5】

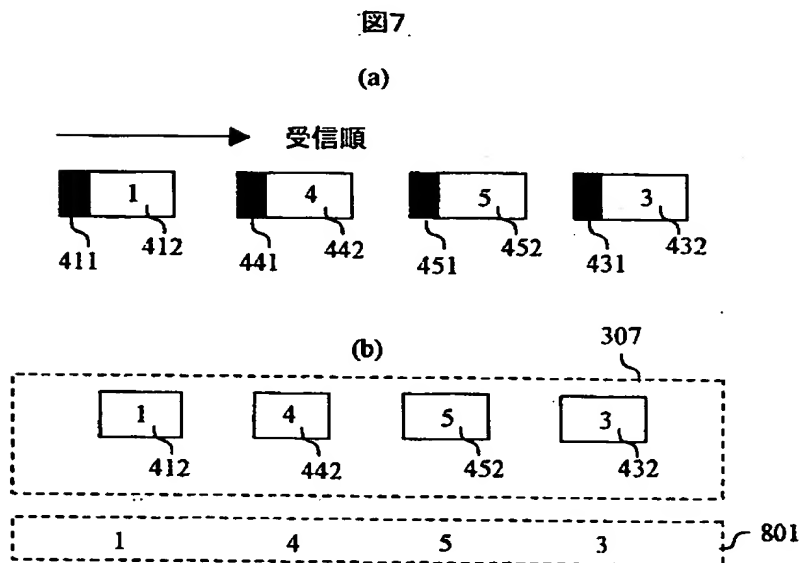
図5



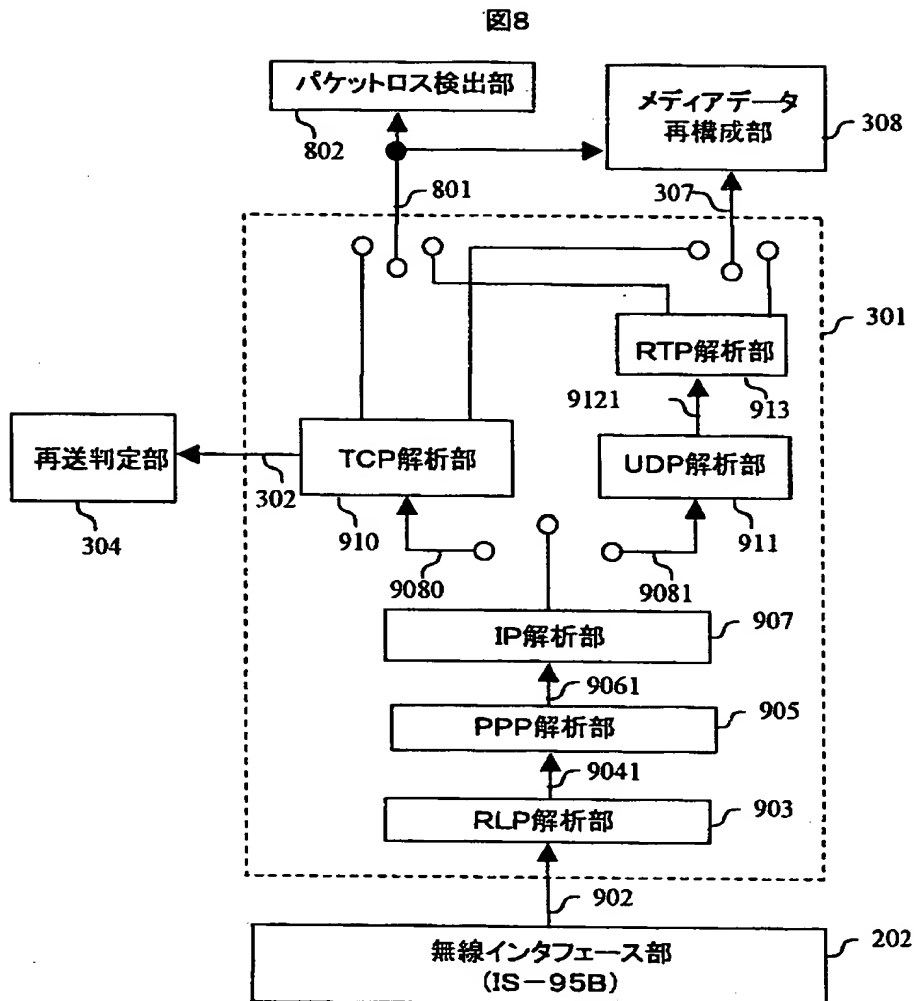
【図 6】



【図 7】



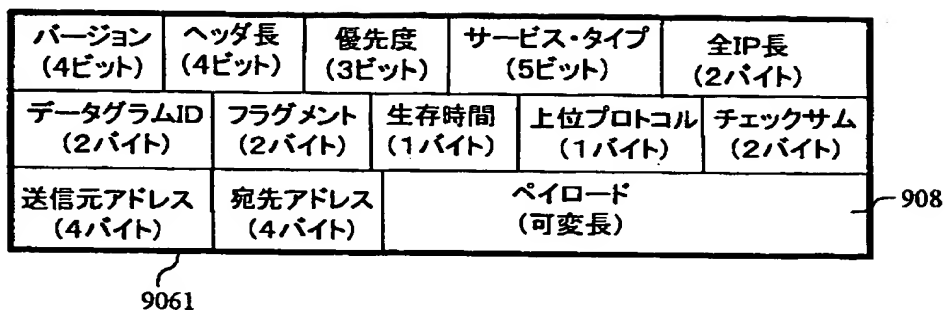
【図 8】



【図 9】

図9

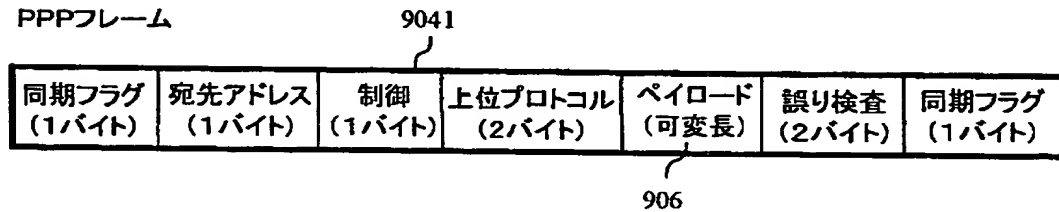
IPv4パケット



【図 10】

図10

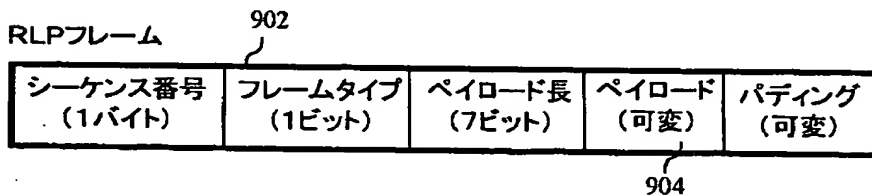
PPPフレーム



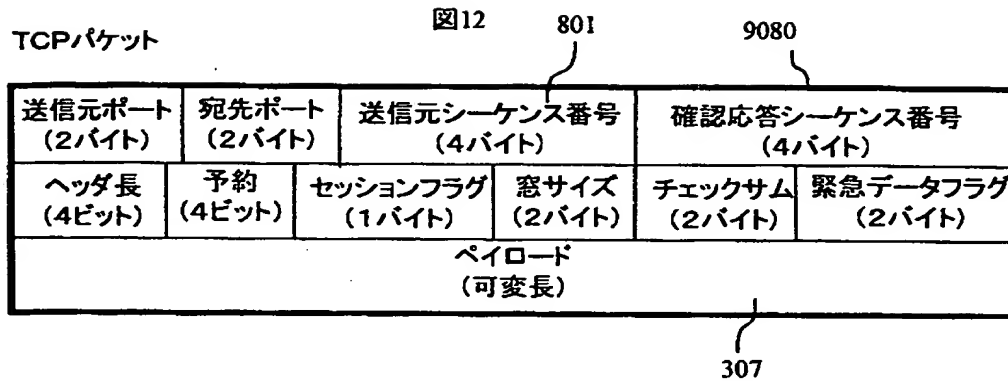
【図 11】

図11

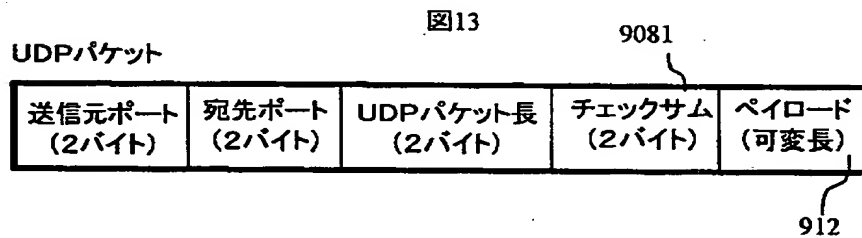
RLPフレーム



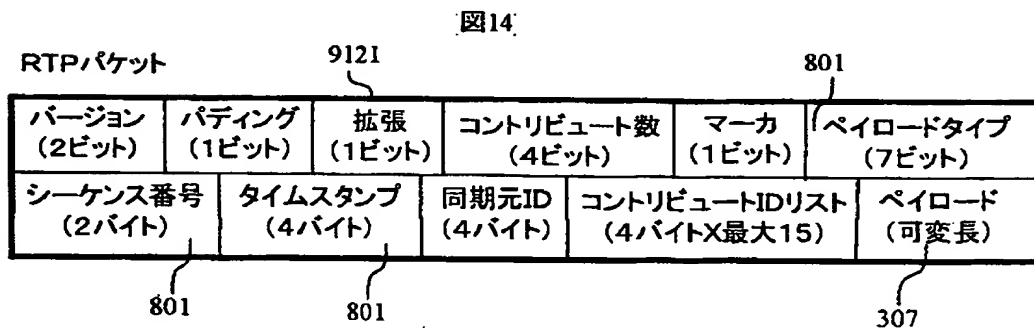
【図 1 2】



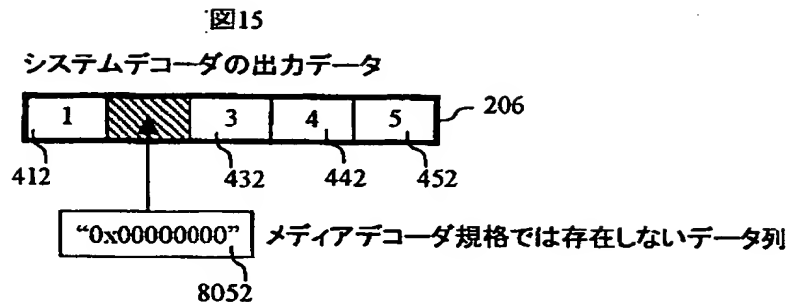
【図 1 3】



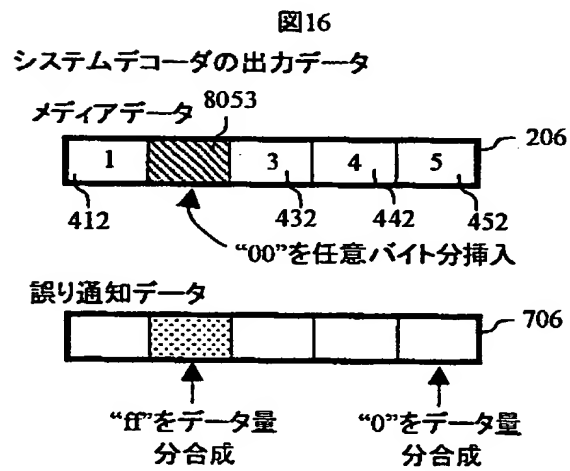
【図 1 4】



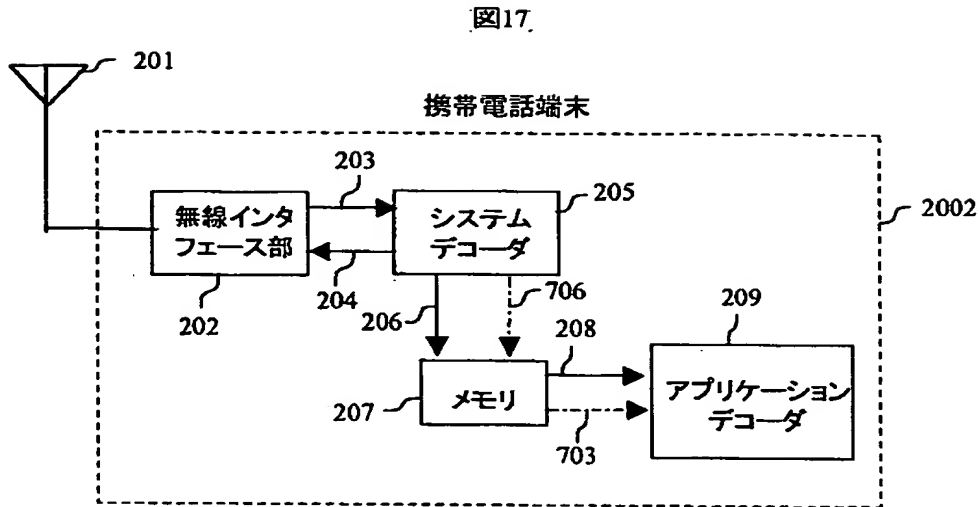
【図 1 5】



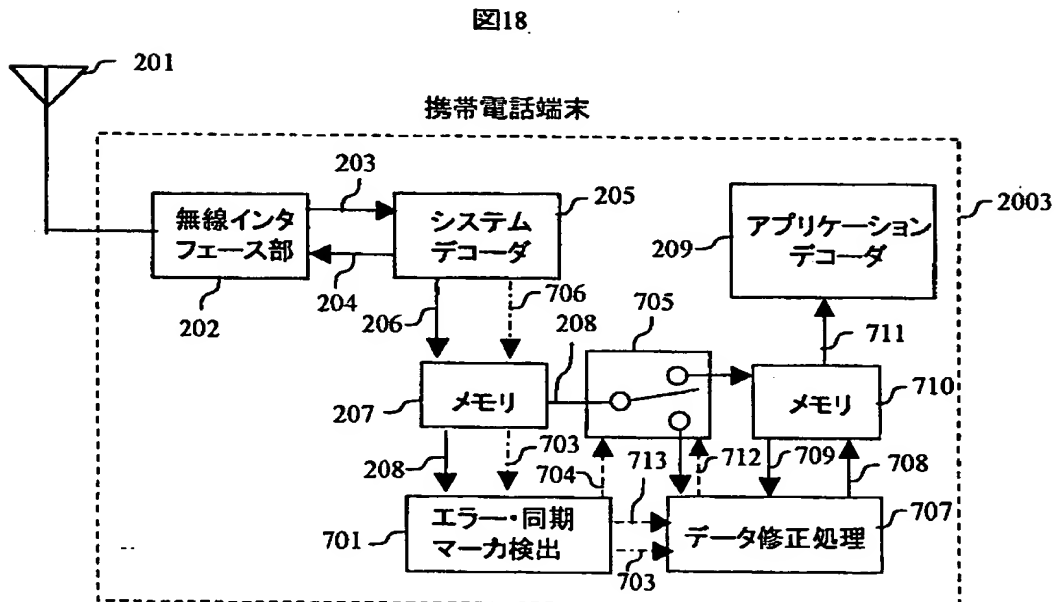
【図 1 6】



【図 17】

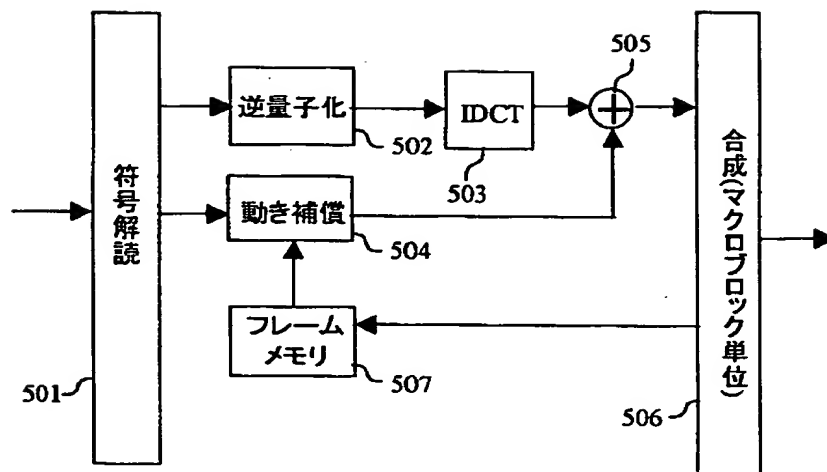


【図 18】



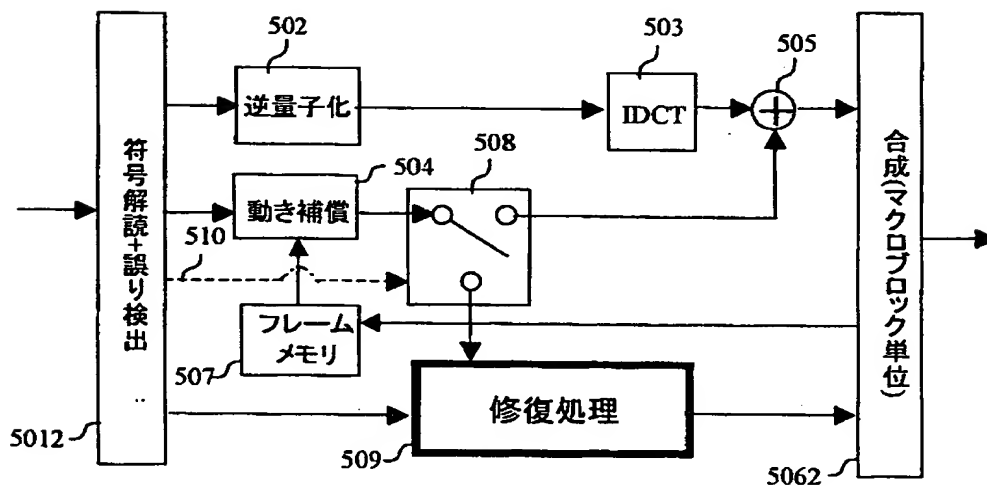
【図19】

図19

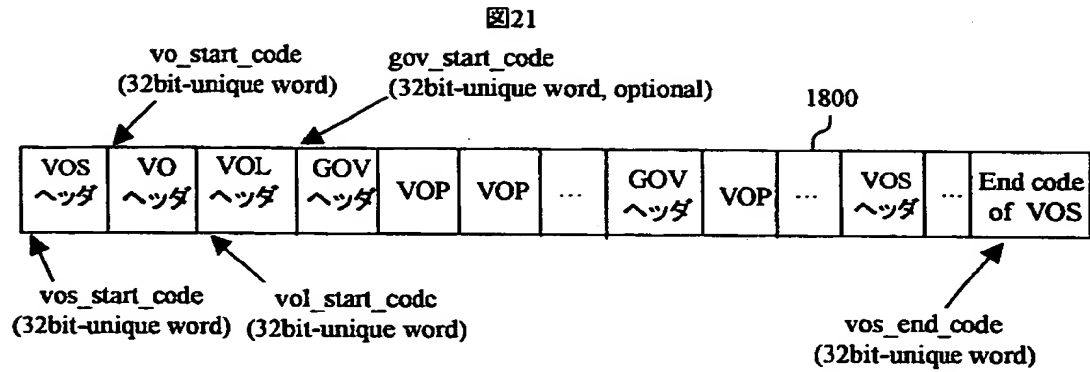


【図20】

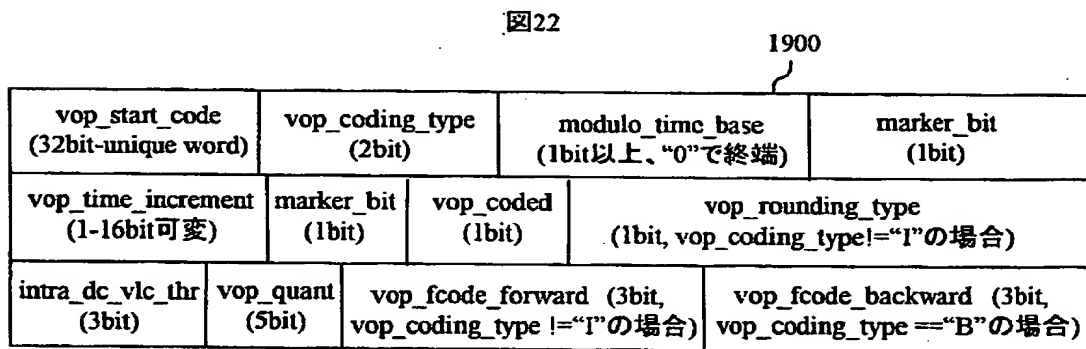
図20



【図 2 1】

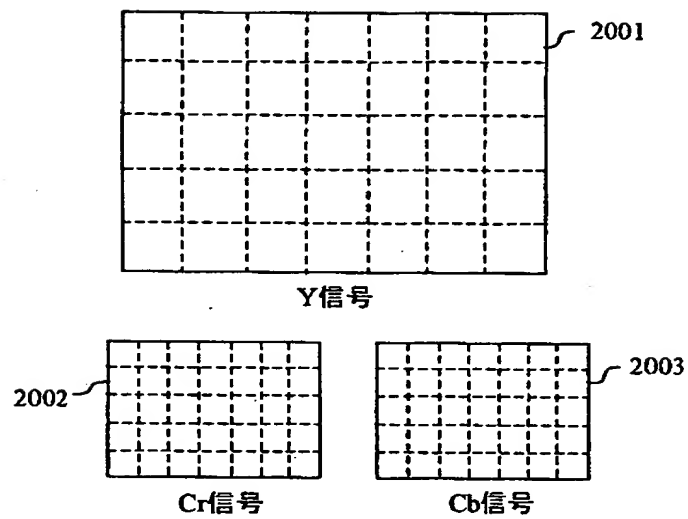


【図 2 2】



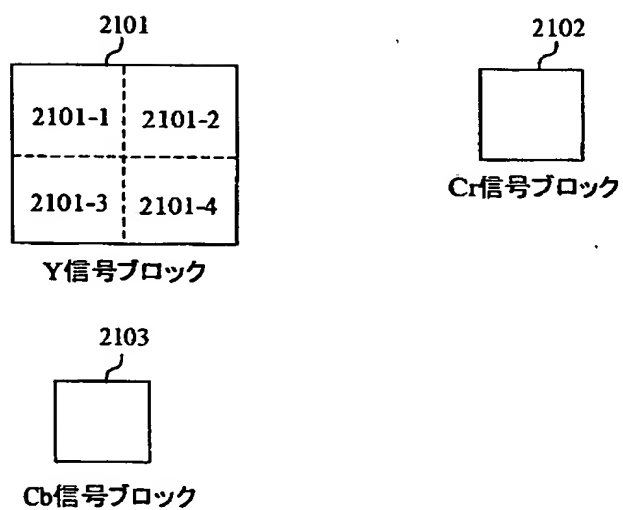
【図 2 3】

図23.

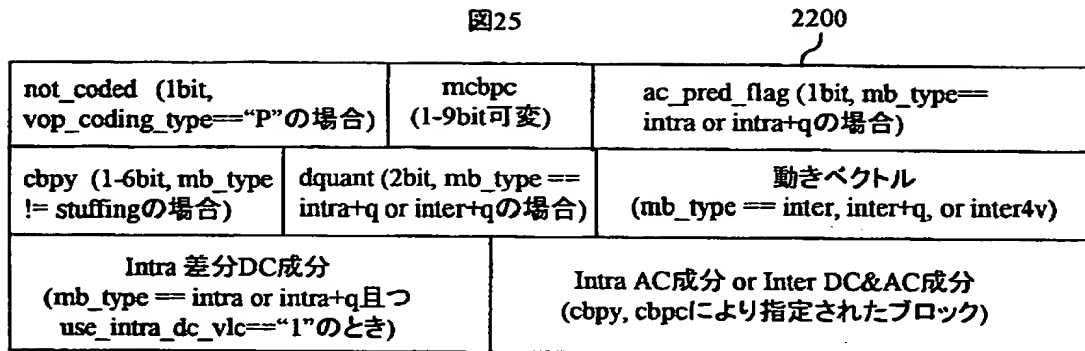


【図 2 4】

図24.

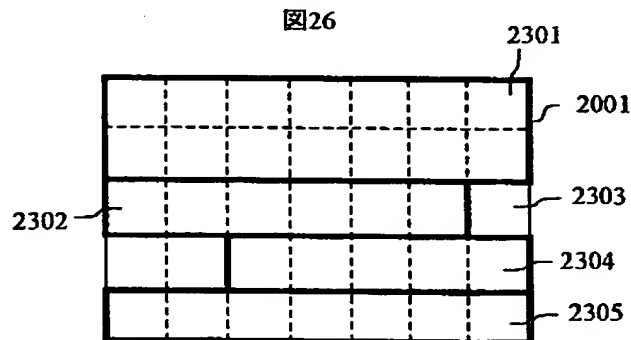


【図 2 5】



mcbpc : mb_type (intra, intra+q, inter, inter+q, inter4v, stuffing), cbpc。
 not_coded : "1"の場合、mb_type=inter, 動きなし、mcbpc以降省略。
 use_intra_dc_vlc: quantとintra_dc_vlc_thrで決定、"0" or "1"の値を取る。

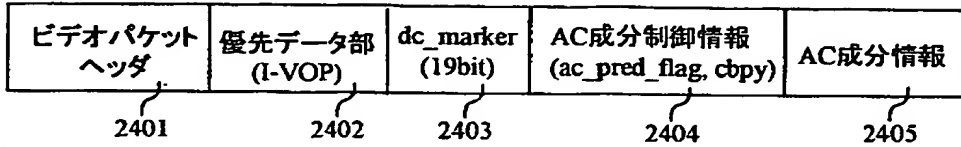
【図 2 6】



【図 2 7】

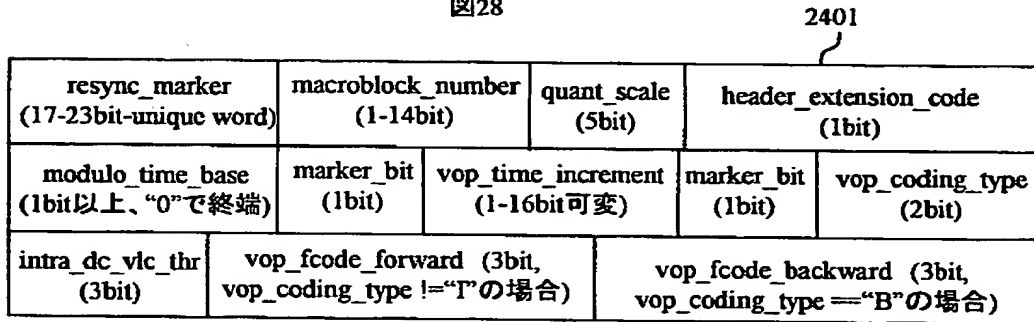
図27

ビデオ packets データ (I-VOP)



【図 2 8】

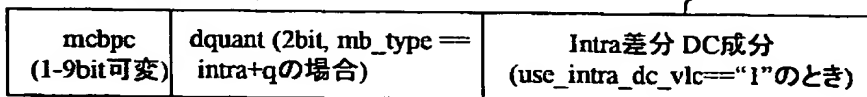
図28



【図 2 9】

図29

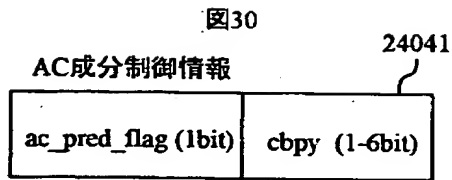
優先データ部 (I-VOP)



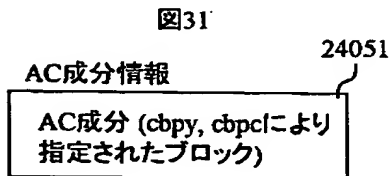
mcbpc : mb_type (intra, intra+q, stuffing), cbpc

use_intra_dc_vlc: quantとintra_dc_vlc_thrで決定、“0” or “1”の値を取る

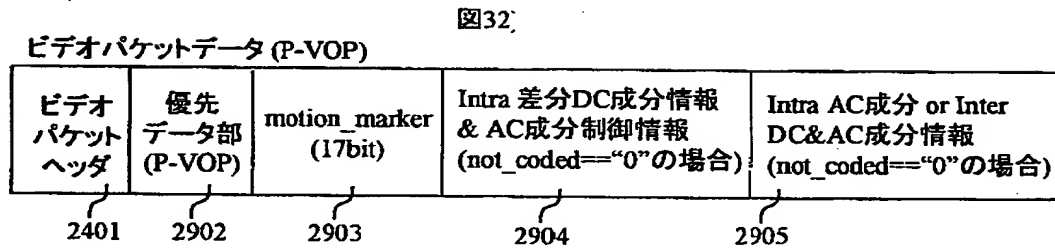
【図 3 0】



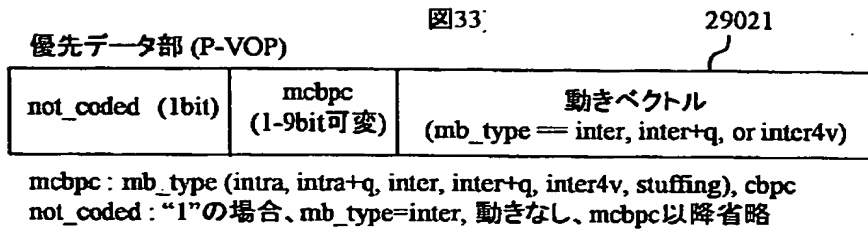
【図 3 1】



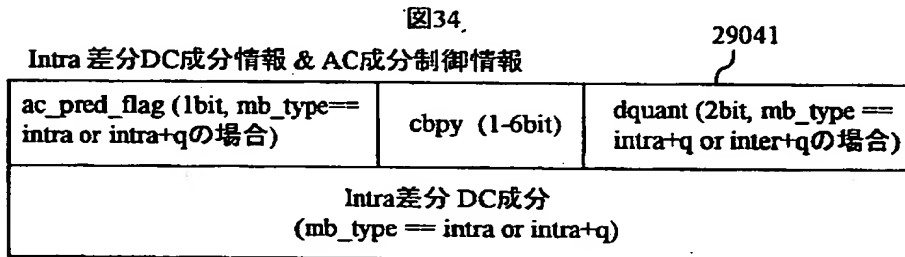
【図 3 2】



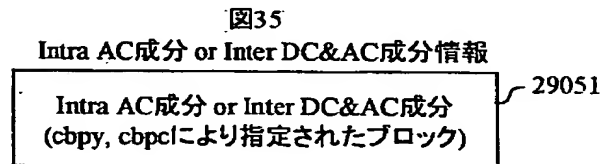
【図 3 3】



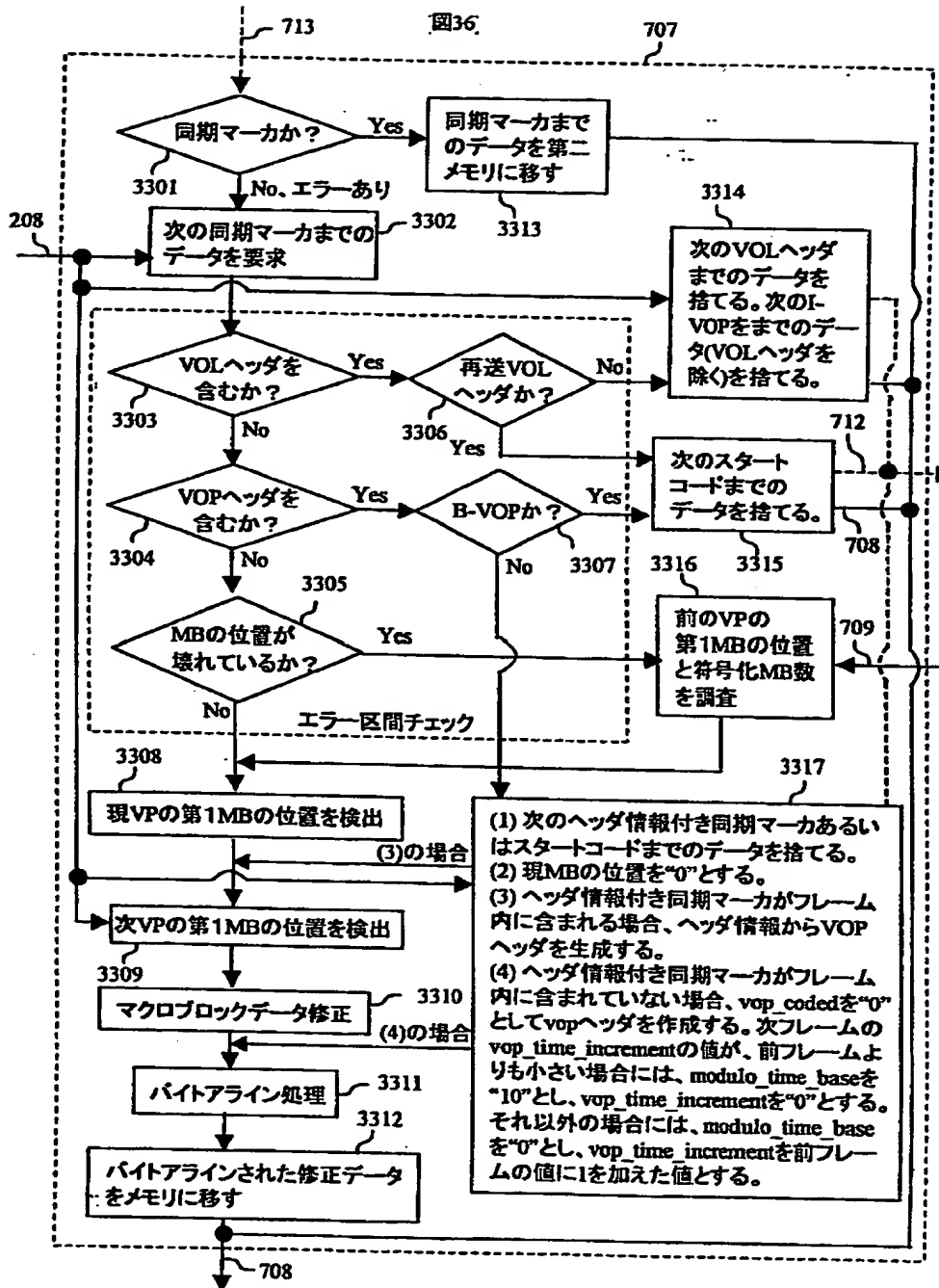
【図 3 4】



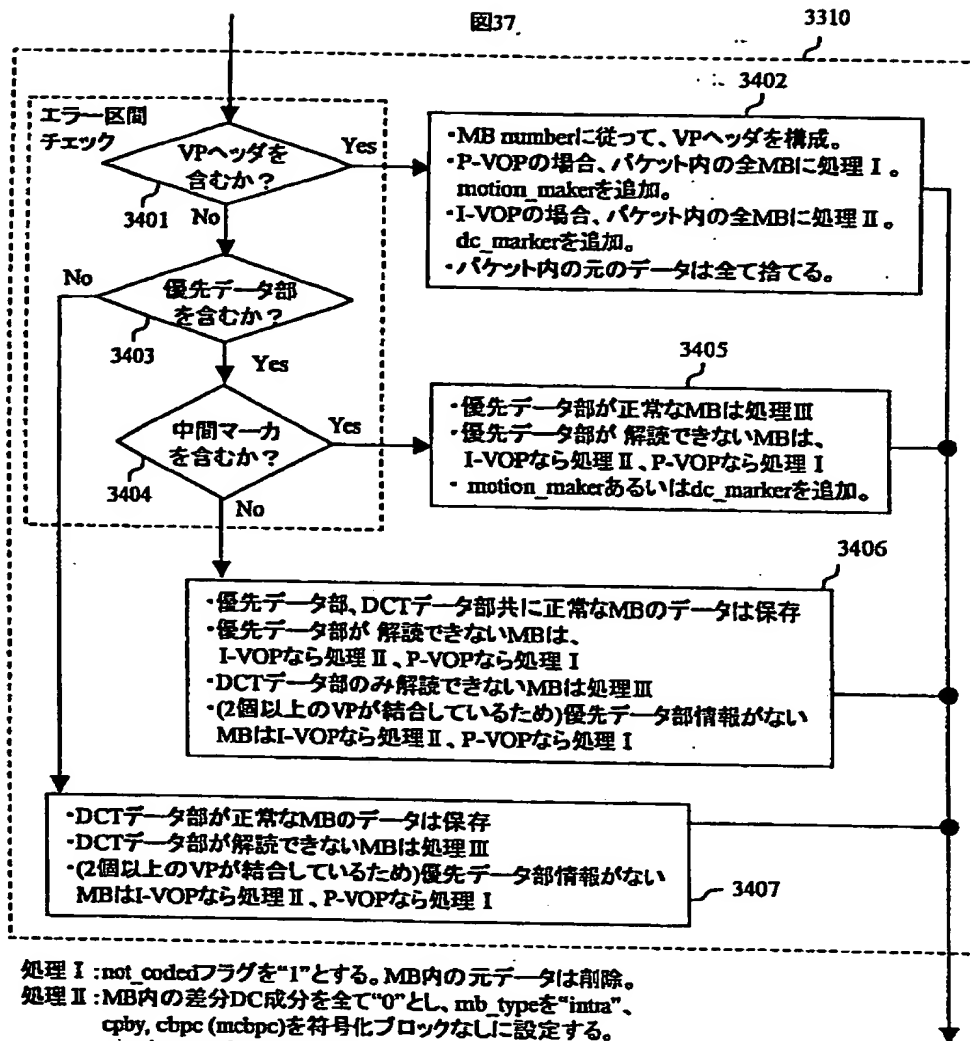
【図 3 5】



【図 36】



【図 3 7】

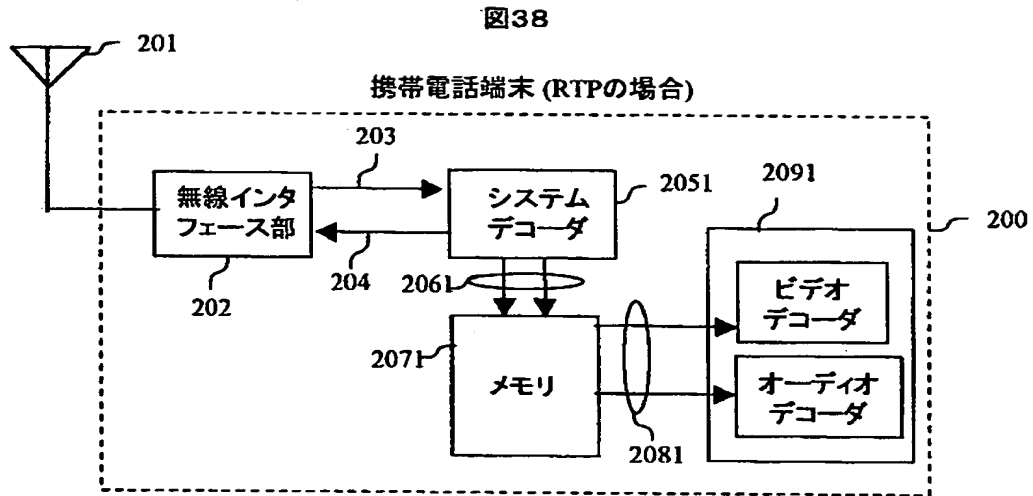


処理 I : not_codedフラグを"1"とする。MB内の元データは削除。

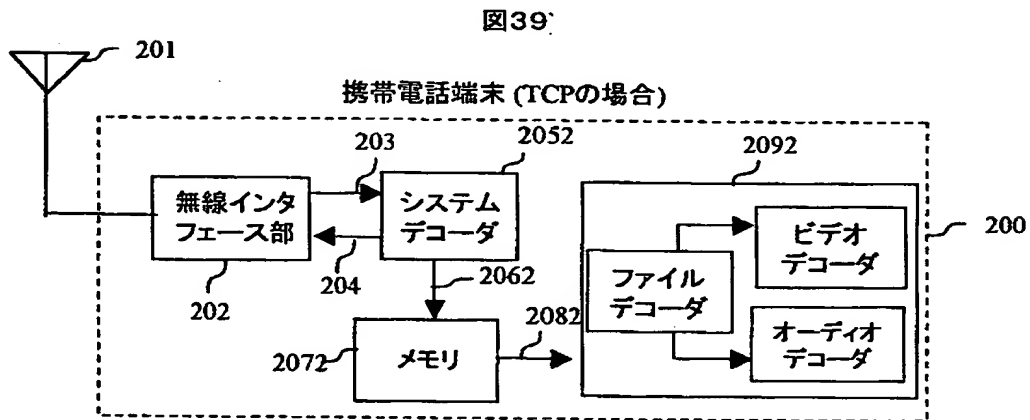
処理 II : MB内の差分DC成分を全て"0"とし、mb_typeを"intra"、
cpby, cbpc (mcbpc)を符号化ブロックなしに設定する。
MB内の元データは削除。

処理 III : cpby, cbpc (mcbpc)を符号化ブロックなしに設定する。さらに、
I-VOPの場合には、ac_pred_flagを"0"とし、AC成分データは削除する。
P-VOPの場合には、mb_typeがイントラ符号化のときは処理 I、
mb_typeが予測符号化のときは、Inter DC&AC成分データを削除する。

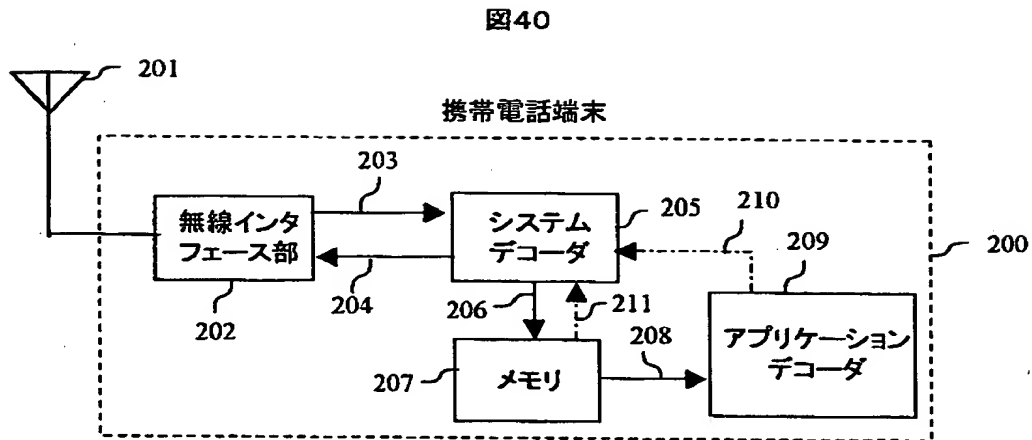
【図 3 8】



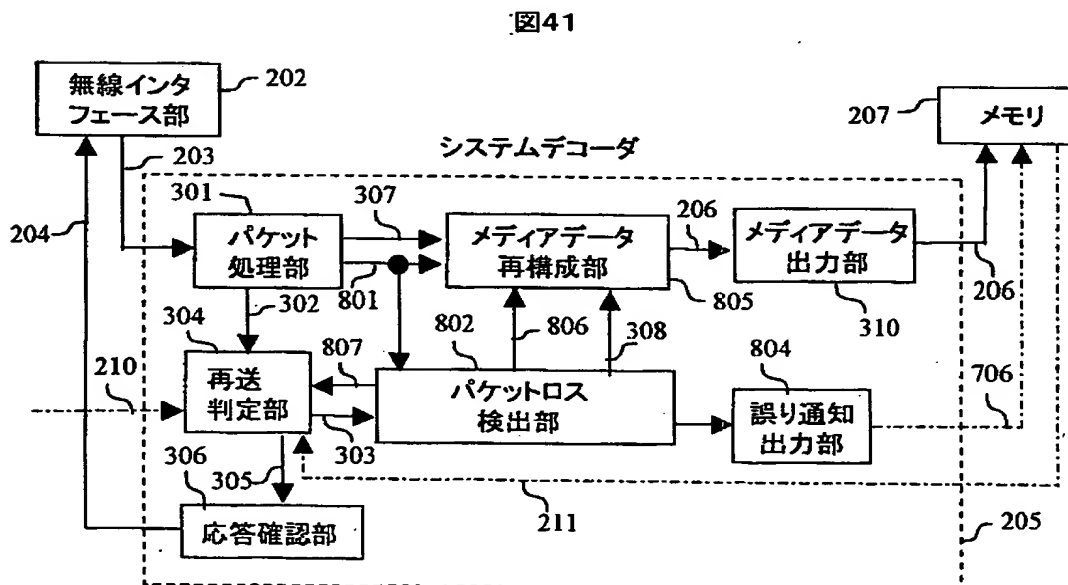
【図 3 9】



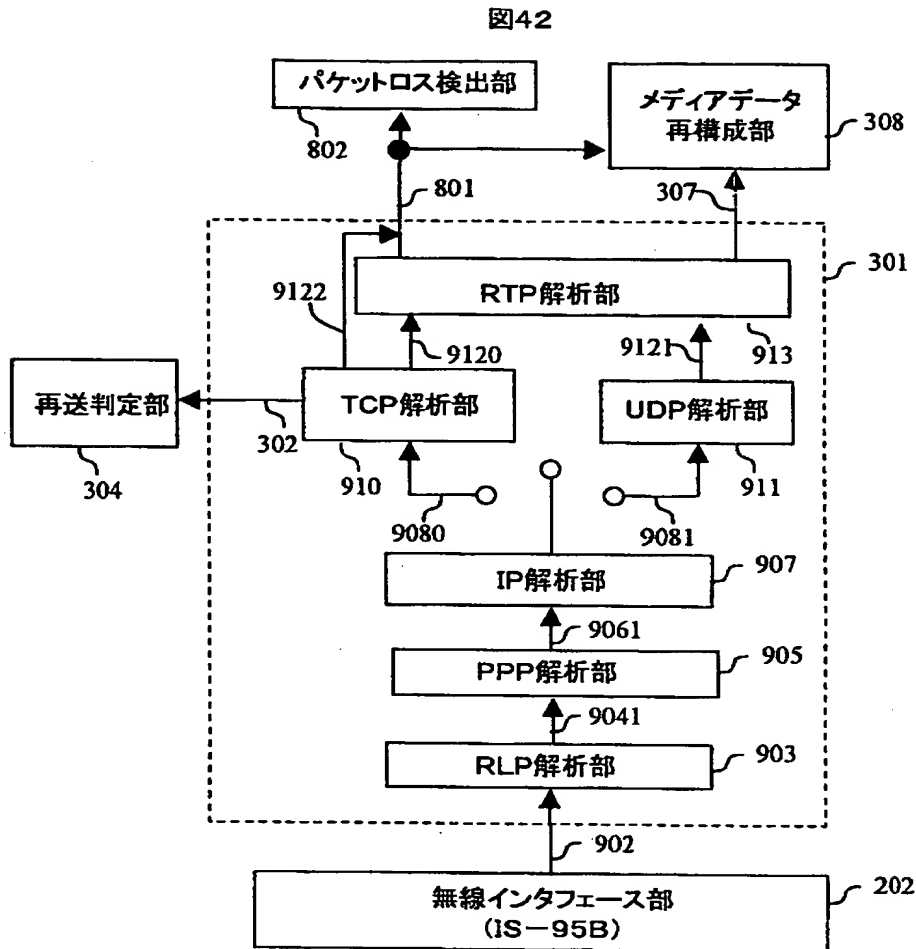
【図 4 0】



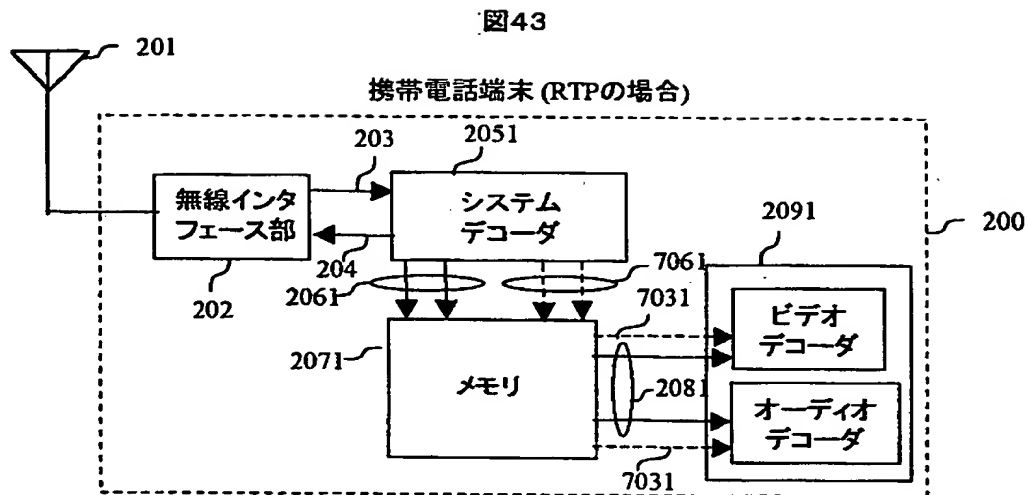
【図 4 1】



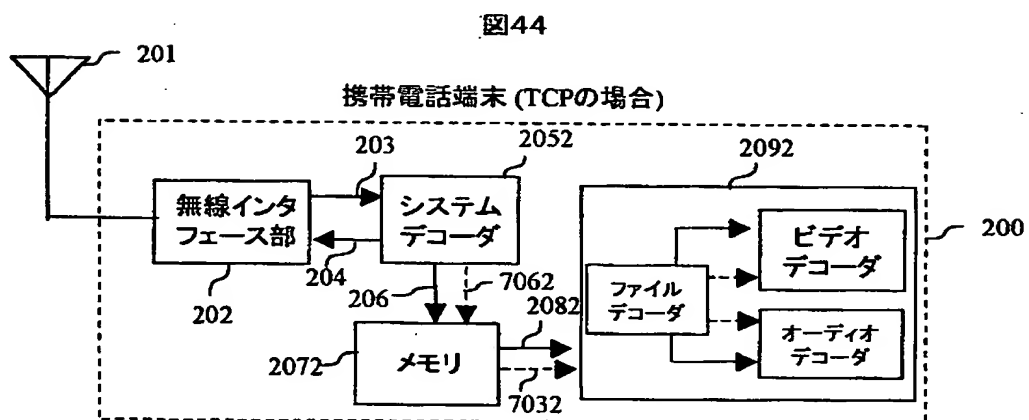
【図42】



【図43】



【図44】



【書類名】 要約書

【要約】

【課題】 デジタルメディアデータの無線通信伝送中にパケットロスが発生した場合に、アプリケーションデコーダに渡されるメディアデータは、ロスが発生したパケットのペイロードを省略した形式となり、伝送誤りの正確な位置を検出することができない。

【解決手段】 受信データのパケットロス部分に、アプリケーションデコーダが明示的に通信エラーが発生したと判定できるアプリケーションデコーダの仕様・規格に含まれないデータ列をシステムデコーダにて埋め込むことにより、アプリケーションデコーダが、伝送誤りの正確な位置を検出することを可能とする。また、埋め込むデータ列をシステムデコーダとアプリケーションデコーダの間で予め決めておくことにより、より正確な誤り位置検出が可能となる。

【選択図】 図 1

認定・付加情報

特許出願の番号	特願2001-208680
受付番号	50101008199
書類名	特許願
担当官	第八担当上席 0097
作成日	平成13年 7月11日

<認定情報・付加情報>

【提出日】	平成13年 7月10日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住 所 東京都千代田区神田駿河台4丁目6番地
氏 名 株式会社日立製作所